



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's Reports

2016

Experimental Investigations of Fused Filament Fabrication for Applications to Affordable Scientific Hardware

Bas Wijnen


Michigan Technological University, bwijnen@mtu.edu

Copyright 2016 Bas Wijnen

Recommended Citation

Wijnen, Bas, "Experimental Investigations of Fused Filament Fabrication for Applications to Affordable Scientific Hardware", Open Access Dissertation, Michigan Technological University, 2016.
<https://digitalcommons.mtu.edu/etdr/222>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etdr>

 Part of the [Polymer and Organic Materials Commons](#)

EXPERIMENTAL INVESTIGATIONS OF FUSED FILAMENT FABRICATION
FOR APPLICATIONS TO AFFORDABLE SCIENTIFIC HARDWARE

By

Bas Wijnen

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Materials Science and Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2016

Copyright 2016 Bas Wijnen

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Materials Science and Engineering.

Department of Materials Science and Engineering

Dissertation Advisor: *Dr. Joshua M. Pearce*

Committee Member: *Dr. Yongmei M. Jin*

Committee Member: *Dr. Paul G. Sanders*

Committee Member: *Dr. Chelsea Schelly*

Department Chair: *Dr. Stephen L. Kampe*

Contents

List of Figures	ix
List of Tables	xi
Preface	xiii
Acknowledgments	xv
Abstract	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Outline of the Dissertation	2
1.2.1 Printed Scientific Tools	2
1.2.2 Modifying the Printer	2
1.2.3 Improving the Printing Process	3
2 Free and Open-Source Control Software for 3-D Motion and Processing	5
2.1 Introduction	6
2.2 Comparison of Franklin with Other RepRap Firmwares	8
2.3 Implementation and Architecture	9
2.3.1 System Description	9
2.3.2 Franklin Firmware	10
2.3.2.1 Temperature Controls	11
2.3.2.2 GPIO Controls	11
2.3.2.3 Stepper Motor Control	12
2.3.3 Franklin C Driver	12
2.3.4 Franklin Python Driver	13
2.3.5 Franklin Server	13
2.3.6 Communication	14
2.3.7 Controlling Movements	20
2.3.8 Probing	21
2.3.9 Scripting	23
2.4 Quality Control	23

2.5	Operating System	25
2.6	Additional System Requirements	25
2.7	Dependencies	25
2.8	List of Contributors	26
2.9	Software Location	26
2.9.1	Archive	26
2.9.2	Code Repository	26
2.9.3	Language	26
2.10	Conclusions	28
2.11	Competing Interests	28
2.12	Acknowledgements	28
3	Open-Source Mobile Water Quality Testing Platform	29
3.1	Methods	31
3.1.1	Design Methodology	31
3.1.2	Performance measurement methodology	33
3.2	Results and Discussion	33
3.3	Limitations and Future Work	35
3.4	Conclusions	36
3.5	Acknowledgements	37
4	Open-source Syringe Pump Library	39
4.1	Introduction	40
4.2	Materials and Methods	41
4.3	OpenSCAD and Open-source 3-D Printing	41
4.4	Syringe Pump Control and Interface	44
4.5	Calibration and Performance Assessment Methodology	44
4.6	Results	46
4.7	Discussion	51
4.8	Conclusions	52
4.9	Acknowledgments	52
5	Free and Open Source Automated 3-D Microscope	53
5.1	Introduction	54
5.2	Background	55
5.3	Experimental	56
5.4	Results and Discussion	57
5.4.1	Results	57
5.4.2	Discussion	62
5.5	Conclusions	65
5.6	Acknowledgments	66

6	Improved Model and Experimental Validation of Deformation in Fused Filament Fabrication of Poly Lactic Acid	69
6.1	Introduction	71
6.2	Background	72
6.2.1	Models of Deformation in FFF	72
6.2.2	PLA and FFF	73
6.3	Methods	74
6.3.1	Materials	74
6.3.2	3-D printing	74
6.3.3	Measurements	75
6.3.3.1	Temperature	75
6.3.3.2	Temperature Model	76
6.3.3.3	Crystallinity	79
6.4	Results	80
6.5	Discussion	83
6.6	Conclusions	88
7	Future Work	89
7.1	3-D Printer Improvements	89
7.2	Software Improvements	90
7.3	Education	90
7.4	Deformation	91
	References	93
A	Franklin Source Statistics	109

List of Figures

2.1	Schematic of the workflow of creating a 3-D printed object using Franklin.	9
2.2	Franklin web interface.	14
2.3	Schematic representation of how the move comprising a segment is prepared in Franklin.	22
3.1	Schematic of the OS mobile water quality testing platform.	32
3.2	OS water testing platform with the assembled case, electronics and an inset schematic of case design in OpenSCAD.	32
3.3	Calibration of the OS nephelometer. The line shown is the result of the calibration.	34
3.4	Comparison of the OS nephelometer and the Hach 2100P. Points which have identical readings on both devices are on the line.	35
4.1	3-D printable parts for the open-source syringe pump.	42
4.2	Electronic schematic of Open-Source Syringe Pump.	45
4.3	Screenshot of Syringe Pump Web Interface.	46
4.4	Exploded view of Open-Source Syringe Pump.	47
4.5	Digital Photograph of Open-Source Syringe Pump version Nema 11.	47
4.6	Digital Photograph of Open-Source Syringe Pump version Nema 17.	47
4.7	Digital Photograph of Open-Source Syringe Pump version of the Dual Nema 17 Pump.	48
4.8	Digital Photograph of the Dual Pump connection.	48
5.1	The open-source 3-D microscope stage with USB microscope.	58
5.2	A) OpenSCAD STL rendering of USB microscope holder, B) Assembled USB microscope holder with ball bearings and screws.	59
5.3	The 3-D microscope stage shown adapted to a stage. The stage is viewed from above by a conventional microscope.	60
5.4	The electronic schematic of the 3-D microscope.	61
5.5	The Franklin interface allows users to control the movement of the USB microscope in x, y and z directions. Furthermore, G-Code files can be uploaded and executed.	62

5.6	a) An image of a flower where the bottom is in focus. b) An image of a flower where the top is in focus. c) The final image of the stacking technique results in an image of greater depth than any of the original images.	63
5.7	Final photostitched image of complex aggregate paint.	67
6.1	Commonly observed deformation in the first layers of 3-D printed PLA.	72
6.2	A representative infrared images that was used to measure the temperature of the printed object.	76
6.3	Diagram showing the meaning of the variables in an exaggerated example of warping.	79
6.4	Samples that were used for the XRD measurements. a) Photograph of the sample, dimensions are 10 cm × 2.8 cm × 1 cm. b) Diagram of the print path for odd layers. c) Diagram of the print path for even layers. The number of lines in the diagrams of the print paths has been reduced for clarity.	80
6.5	Vertical gradient of the temperature at several points in time. The curves are the fit of the thermal model to the data.	82
6.6	XRD spectra for the sample that was only annealed at 50°C, and the sample that was first annealed for two hours at 50°C, then for two hours at 90°C.	83
6.7	Crystallinity as a function of annealing time when annealing at 90°C.	84
6.8	Changes in size after annealing at 50°C and 90°C. At 50°C (blue markers) there is no change in size, while at 90°C (red markers) the objects have shrunk horizontally, but grown vertically.	85
6.9	The surface of the fracture interface of 1 hour annealed (top) and unannealed (bottom) samples, taken with an Environmental Electron Scanning Microscope (ESEM).	86
6.10	Measured warp and model predictions (Equation 6.5). The curve is the model's prediction with the variables from the fit to the thermal graph.	87

List of Tables

2.1	Current RepRap Firmware.	6
2.2	Supported functions.	16
2.3	Supported functions. (Continued)	17
2.4	Commands for the Franklin C Driver.	18
2.5	Commands for the Franklin C Driver. (Continued)	19
2.6	Supported Commands, Function and Response.	20
4.1	Bill of Materials for three examples of the open-source syringe pumps. All parts are printed in PLA with 20% infill. The cost of a dual NEMA17 pump is twice that of a single NEMA17 pump, except only one Raspberry Pi is needed.	43
4.2	Coefficient of variation as a function of microstepping for NEMA 11 and NEMA 17 open-source syringe pumps.	49
4.3	Specifications for the open-source syringe pump are shown compared to commercial pumps.	50
5.1	Comparison of open source microscope to proprietary microscope specifications.	64
6.1	Selected properties of the polymer that was used during the experiments.	75

Preface

This thesis is composed of five published (or submitted for publishing) papers. The authors' contributions are described hereafter.

Chapter 2 is

B. Wijnen, G. C. Anzalone, A. S. Haselhuhn, P. Sanders, and J. M. Pearce, "Free and open-source control software for 3-d motion and processing," *Journal of Open Research Software*, vol. 4, no. 1, 2016.

B. Wijnen's contribution to this work involved writing Franklin and writing the manuscript.

G. C. Anzalone's contribution to this work involved assistance with the experiments and editing the manuscript.

A. S. Haselhuhn's contribution to this work involved performing the experiments and helping to write and edit the manuscript.

P. Sanders's contribution to this work involved assistance with the experiments and editing the manuscript.

J. M. Pearce contributed on writing, experimental design, editing and consultation.

Chapter 3 is

B. Wijnen, G. Anzalone, and J. M. Pearce, "Open-source mobile water quality testing platform," *Journal of Water Sanitation and Hygiene for Development*, vol. 4, no. 3, pp. 532–537, 2014.

B. Wijnen's contribution to this work involved writing the control software for the device, design, build and test it and helping to write and edit the manuscript.

G. Anzalone's contribution to this work involved designing, building and testing the device and writing and editing the manuscript.

J. M. Pearce contributed on writing, experimental design, editing and consultation.

Chapter 4 is

B. Wijnen, E. J. Hunt, G. C. Anzalone, and J. M. Pearce, "Open-source syringe pump library," *PloS one*, vol. 9, no. 9, p. e107216, 2014.

B. Wijnen's contribution to this work involved writing the control software for the device, design, build and test it and helping to write and edit the manuscript.

E. J. Hunt's contribution to this work involved writing the manuscript and testing the device.

G. C. Anzalone's contribution to this work involved designing and building the device, and assisting with the experiments.

J. M. Pearce contributed on writing, experimental design, editing and consultation.

Chapter 5 has been accepted and will soon be published as:

B. Wijnen, E. E. Petersen, E. J. Hunt, and J. M. Pearce, “Free and Open Source Automated 3-D Microscope,” *Journal of Microscopy*, (accepted for publication), 2016.

B. Wijnen’s contribution to this work involved writing the control software for the device, design, build and test it and helping to write and edit the manuscript.

E. E. Petersen’s contribution to this work involved testing the device and editing the manuscript.

E. J. Hunt’s contribution to this work involved editing the manuscript.

J. M. Pearce contributed on writing, experimental design, editing and consultation.

Chapter 6 is

B. Wijnen, P. G. Sanders, and J. M. Pearce, “Improved Model and Experimental Validation of Deformation in Fused Filament Fabrication of Poly Lactic Acid,” (*to be published*), 2016.

B. Wijnen’s contribution to this work involved designing and performing the experiments and analysis and writing and editing the manuscript.

P. G. Sanders contributed on writing, experimental design, editing and consultation.

J. M. Pearce contributed on writing, experimental design, editing and consultation.

Acknowledgments

I thank Dr. Pearce for his efforts to make it possible for me to work with him, and for his help with everything here.

I thank Dr. Paul, Dr. Jin and Dr. Schelly as the members of my committee for their time to help me get through the last part of this project and for their assistance.

I thank all the members of the MOST group for the pleasant time working together.

I thank the friends I have made here for spending their time with me. I am not going to list all of them. Just the most important: thank you Jerry and Joy, you have been like family to me and I appreciate that a lot. We will keep in touch, and hopefully we can soon meet in the Netherlands.

And I thank Aeva, for everything. You are a wonderful person, and I feel honored to be your friend.

Finally for the people I met here, I thank Jephias and David. It has been a pleasure living with you.

I also want to thank some people I didn't meet, for keeping me informed and entertained with much less bias than most other news sources in this country: I thank Jon Stewart, John Oliver and all of The Young Turks for that.

Daarmee ben ik aangekomen bij de achterblijvers in Nederland. Allereerst mijn ouders, dank jullie wel voor de steun en hulp om mijn reis voor elkaar te krijgen, tijdens mijn verblijf hier en mijn bezoeken aan Nederland. Ik weet dat vooral mamma het moeilijk vond dat ik zo ver weg ging, dank je wel dat je me ondanks dat steunde in mijn keus.

Ook de andere achterblijvers in Nederland wil ik bedanken. Degenen die de tijd namen om mij te zien tijdens mijn bezoeken aan Nederland, en degenen die ik graag bezocht had maar waarbij dat er niet van gekomen is: Loes, Rien, Dieneke, Agnes, de hele familie Wijnen en de hele familie van den Brink, Inge, Saskia, Mark, Maaïke en Reinout, Bart en Sander, Mark en Enti, Sybren, Jan, en Adriaan. En natuurlijk de Egel, ik bedoel Machteld.

Verder wil ik de galanten bedanken dat ik daar altijd welkom ben, en in het bijzonder Kasper en Marieke voor hun steun aan mij en Wieke.

En ten slotte natuurlijk Wieke. Dank je wel voor je steun en het contact dat we hadden in deze tijd. Het is niet makkelijk om een relatie te onderhouden over zo'n afstand, en binnenkort hoeft dat gelukkig niet meer. Dank je voor je steun en geduld.

Abstract

This research aims to make science more accessible through the use of open source 3-D printers.

A new control system for CNC machines in general and 3-D printers specifically, is developed and presented that enables web-based control of 3-D tools from any Internet connected device. The system can be set up and controlled entirely from a web interface.

A tool was designed to help apply science to the developing world. The developing world remains plagued by lack of access to safe drinking water. A methodology is provided for the design, development, and technical validation of a low-cost open-source water testing platform. A case study is presented where the platform is developed to provide both colorimetry and nephelometry. This approach resulted in equipment that costs between 7.5 and 15 times less than current commercially available tools.

A syringe pump was designed and manufactured using an open-source hardware and software. The design, bill of materials and assembly instructions are available. The cost of the entire system, including the controller and web-based control interface, is on the order of 5% or less of a commercial syringe pump having similar performance. The design should suit the needs of almost any research activity requiring a syringe pump.

A low-cost, open source 3-D microscope stage is presented. A RepRap 3-D printer was converted into an optical microscope equipped with a customized holder for a USB microscope. The machine is able to operate with USB or conventional microscopes. The repeatability is below 2-D microscope stages, but it is adequate for the majority of scientific applications. The stage costs less than 3% to 9% of the closest proprietary commercial stages.

A deformation model for PLA was expanded to use a physics based temperature gradient. This generalized the model to 3-D printing in a room temperature environment. Tests confirm that this is a valid model for predicting warpage of thin vertical walls of PLA. Additionally, the effect of annealing was examined. It was found that at a temperature of 50°C, no shrinkage or crystallization takes place, but at 90°C the plastic rapidly crystallizes to around 20% crystallinity.

Chapter 1

Introduction

1.1 Motivation

An important difference between humans and most other animals is our high level of intelligence. Throughout history, while most animals survive and procreate by adapting their body to the environment through evolution, humans have in many cases used their mind to adapt their environment to their body.

It should therefore be no surprise that humans seem to have an intrinsic hunger for knowledge. But there are limits to how much a person can learn. For many people, those limits are not rooted in their physical abilities, but in the society they live in. Learning costs time, and often money. While for some people it is possible to be paid for learning, many do not have that luxury, or at the very least they get a limited budget.

The modern scientific community publishes its findings for the world to see. This greatly increases the pace at which new discoveries are made, because it allows scientists to avoid reinventing things. A major limitation that remains is the amount of people working on it.

By making scientific tools more affordable, more people can join the scientific community and people in the community can be more productive. The result of this is not just increased happiness for those people. It also allows them to use their new knowledge to improve living conditions for themselves and the people around them.

1.2 Outline of the Dissertation

This dissertation focuses on making science more accessible through the use of 3-D printers. Low cost 3-D printers like the RepRap (self Replicating Rapid prototyper) are relatively new, and they have a lot of potential for accomplishing this.

The way they were used until now was limited, however. In order to make them more flexible, new control software was needed. To overcome this limitation, an open source 3-D machine control software was designed and implemented for this purpose. It is described in Chapter 2. This allowed 3-D printers to be used for the projects described in the latter chapters.

1.2.1 Printed Scientific Tools

An example of technology that can improve living conditions is described in Chapter 3: a low cost nephelometer, which can be used for water quality measurements. This 3-D printed device not only allows people in the developing world to join the scientific community by using this device for measurements in their region, but it also allows them to use the device for preventing sickness.

Similarly, Chapter 4 describes an open source 3-D printable syringe pump, which allows disadvantaged people to use such a device for scientific research or any other purpose (e.g. medicine), and people who could previously afford syringe pumps can now spend more money on other costs, so it improves their efficiency as well.

1.2.2 Modifying the Printer

Printed tools such as described above are very useful, but the printer can do more. It is itself a robot which can be modified to be used as a more general purpose scientific instrument. For example, the syringe pump designed and discussed in Chapter 4 was slightly modified and mounted on a 3-D printer to print liquids. In Chapter 5, the process is described to modify a 3-D printer to mount a USB microscope instead of a print tool. Modifications like this increase the value of the 3-D printer, both because they allow it to be used for scientific methods (microscopy, in this case), but also because the printer and microscope now take as much space together as one of them

did before.

1.2.3 Improving the Printing Process

As described above, RepRap 3-D printers in their current state are very capable scientific tools. But there is room for improvement. The printed parts deform during and after printing, and the physics behind this have not been explained well. In Chapter 6 the mechanism behind the deformation is evaluated to improve an existing model based on careful thermal measurements. The insights that this provides can help to prevent or compensate for deformation in the future, which would allow more precise and larger objects to be printed using the fused filament fabrication method that is used by RepRap 3-D printers.

Chapter 2

Free and Open-Source Control Software for 3-D Motion and Processing

abstract¹

RepRap 3-D printers and their derivatives using conventional firmware are limited by: 1) requiring technical knowledge, 2) poor resilience with unreliable hardware, and 3) poor integration in complicated systems. In this paper, a new control system called Franklin, for CNC machines in general and 3-D printers specifically, is presented that enables web-based three dimensional control of additive, subtractive and analytical tools from any Internet connected device. Franklin can be set up and controlled entirely from a web interface; it uses a custom protocol which allows it to continue printing when the connection is temporarily lost, and allows communication with scripts.

¹The material contained in this chapter was previously published in the Journal of Open Research Software[1]

(1) Overview

2.1 Introduction

Since Bowyer’s open source release of the self-replicating rapid prototyper (RepRap) [2, 3], people from all over the world have been building and improving RepRap 3-D printers [4]. The existing control system consists of two parts: firmware on the 3-D printer to send the signals through to individual components, and a program on a desktop computer to send instructions (G-code) to the firmware. Current firmware options (Table 2.1) use the same interface with the host, and suffer from two fundamental problems: 1) they are not robust and 2) they require a weak microcontroller to do significant work, thus limiting their functionality. Some firmware solve the latter challenge by requiring a more powerful microcontroller, which limits its applicability.

Table 2.1
Current RepRap Firmware.

Name	Hardware	Comments
RepRap Firmware (original)	ARM	
LinuxCNC	PC	Uses the parallel port, or GPIO pins on a Raspberry Pi or BeagleBone
RepRap Firmware (fork by dc42)	ARM	Based on RepRap Firmware
Marlin	AVR	Adaptation of Grbl for 3-D printers
Repetier	AVR, ARM	Based on Marlin
Teacup	AVR, ARM	Runs on AVRs with low memory, such as Arduino Uno
aprinter	AVR, ARM	
Smoothie	ARM	

As there is a standard communication format between existing firmware and hosts, there are also many different options for host software including Pronterface [5], Cura [6] and OctoPrint [7]. In terms of ease of use, OctoPrint stands out: it provides a web interface to control the printer over a network from any device with a browser. OctoPrint supports monitoring the printer with a webcam, and makes recordings of the print. Another project that is worth noting in detail is Pacemaker [8]. Pacemaker

defines a new protocol that tries to move most computationally intensive tasks from the firmware to the host. This protocol replaces G-code and requires firmware that can handle it. The Pacemaker program itself runs on a host computer, converting G-code into the format understood by the firmware. The format is designed to be backwards compatible, so different firmware and hosts can work together. This only works if new configurations are tested and validated, requiring many users to maintain a unique configuration. At this point, it is unclear whether it will gain the critical mass necessary for success.

Reliability has also been identified as a core challenge for low-cost 3-D printers [9]. For a system to work reliably, data sent between components must not be corrupted and as this is not always avoidable, error checking is required. Unfortunately, ubiquitous computer numerical control (CNC) tools of all types most commonly use G-code, which provides a weak checksum for data sent to the device and no protection for replies. For unreliable connections, the host computer reports a lost serial port, almost immediately followed by a newly discovered port. For all of the current control systems summarized in Table 1, when this occurs mid-print, the print fails. This can only be fixed by changing the protocol between firmware and host software, and then both the firmware and the host software need to be changed to implement the new protocol. Recovering from such connection issues is becoming increasingly important as printers become larger (e.g. Gigabot), work on more sophisticated designs such as scientific [10, 11] and medical equipment [12, 13, 14], and solar-powered printers are used to accelerate sustainable development [15, 16, 17].

CNC machines are powerful general purpose motion controllers that are capable of integrating with a system, for example to perform a constant criterion experiment such as recording the adjustments made to an actuator required to keep a measured value constant. While the hardware is capable, the G-code protocol provides no means for performing such functions. Similarly G-code does not integrate well with other programs, which prevents one click print functionality. To solve these two problems a scripted interface is necessary.

In this paper a new control system called Franklin is presented to be used with CNC machines in general and 3-D printers specifically. It was developed while exploring modifications to RepRap 3-D printers, in order to make them more powerful and more user friendly. It solves the problems identified above and allows CNC machines to be more productive and valuable. Franklin was tested on the following RepRap devices: Mendel RepRap, Delta RepRap, OS laser welder, PCB micromill, and the open source metal 3-D printer. The results are presented and discussed.

2.2 Comparison of Franklin with Other RepRap Firmwares

This section compares Franklin in some detail with other available firmwares. It was added to the dissertation and was not part of the published paper.

Franklin and Teacup are the only firmwares that can run on the relatively weak Arduino Uno. Others require at least an Arduino Mega, and both RepRap Firmwares and Smoothie even require an ARM processor.

Franklin uses its own protocol for communication, which includes a strong checksum for communications to and from the firmware. All others use G-Code, which has a weak checksum for data sent to the firmware, and no checksum for replies. Aside from Franklin, only Repetier has the option of using a custom protocol. This protocol has a stronger checksum than G-Code for data sent to the firmware, but does not add a checksum to the reply.

All other firmwares expect the user to open the code with a programming environment in order to upload it to the microcontroller. Some host software bundles one or more firmwares and allows uploading them from the interface in the same way Franklin does.

Franklin requires the firmware to be the same version as the host software, while other firmwares and host softwares can be independently changed. While this makes Franklin less flexible compared to all other firmwares, it also makes it more robust, because Franklin does not need to handle several different “dialects” of G-Code.

Finally, all other firmwares each have their own format for reporting the temperatures and the state of digital input pins, which means that a program that needs this information is either tied to one specific firmware, or it must understand all their formats. Furthermore, they only send this data on request. This is not usable for a setup where an instrument should respond immediately to a signal. Franklin allows scripts to wait for events and they will be notified when those events occur. The driver does not ask for the current state of the input all the time, instead the firmware notifies the driver about any changes.

2.3 Implementation and Architecture

Franklin’s purpose is to: 1) drive CNC hardware including motors, switches, and a tool head and 2) integrate well with the remaining software tool chain, such as 3-D modeling programs and slicers. Slicers translate a 3-D shape described by an STL file and slice it into consecutive thin layers in the z-direction (vertically) as g-code. The full code of Franklin is available on Github under the GNU Affero General Public License [18].

2.3.1 System Description

Reprap 3-D printers have a dedicated real time control board, which is normally based on the Arduino [19] prototyping platform. When using Franklin, the control board contains Franklin Firmware, which handles low-level control and communicates with a more powerful host computer using a serial interface. On the host computer, Franklin Server provides a web interface that can be used from any device on the same network, including the Internet (Figure 2.1). Franklin uses an encrypted (HTTPS) connection and allows restricting access with a password to prevent unauthorized users from controlling the device.

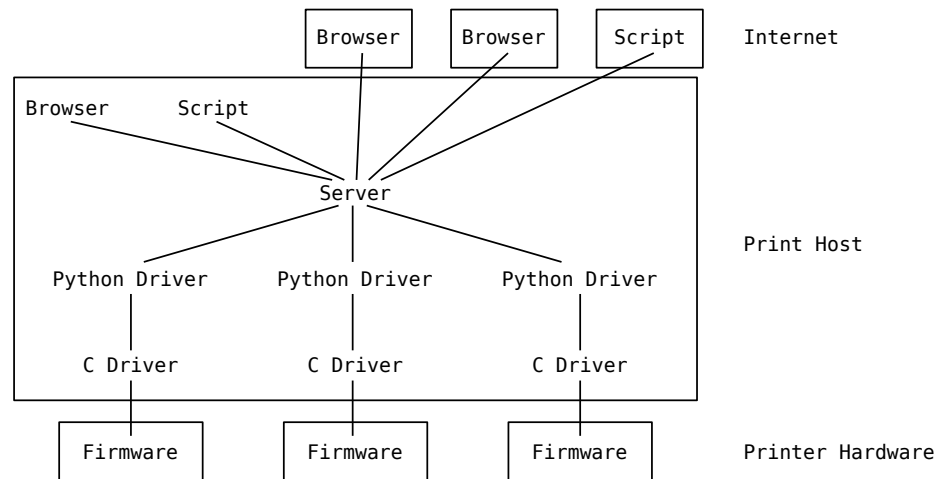


Figure 2.1: Schematic of the workflow of creating a 3-D printed object using Franklin.

The Franklin Server can handle multiple devices simultaneously using a dedicated

driver for each device. This driver consists of two processes: one handling high speed and time sensitive functions written in C++ (the Franklin C Driver), and one handling the rest written in Python (the Franklin Python Driver). In addition to communicating with each other, the Franklin C Driver communicates with the Franklin Firmware, while the Franklin Python Driver communicates with Franklin Server. During device discovery, before the driver is started, the Franklin Server also communicates directly with the Franklin Firmware.

For Franklin to be considered acceptable as a firmware solution, tests were run using the following: Mendel RepRap [20], Delta RepRap [21], quad delta RepRap (4 vertically stacked quad MOST delta RepRap heads run on a single microcontroller and set of three position stepper motors), OS laser welder [11], PCB micromill [22] and the open source metal 3-D printer [23]. The test data obtained from the latter was utilized to further develop the metal 3-D printer design [24]. Rather than including parts produced via fused filament fabrication, the updated 3-D printer design was constructed entirely from metal parts to reduce damage caused by weld splatter. Franklin Firmware and the improved metal 3-D printer were used to establish low-cost substrate release mechanisms that allow metal 3-D printed objects to be removed from a metal print substrate with minimal force [24].

2.3.2 Franklin Firmware

Franklin Firmware only handles tasks that cannot be performed well by the host computer as the microcontroller lacks processing power and memory (8 bit, 16 MHz, no hardware assisted floating point operations) compared to the host computer's processor. Fewer microcontroller tasks ensures more time to complete them, improving overall system performance. Franklin goes further than Pacemaker in this regard. Pacemaker assumes all machines to be Cartesian, and approximates moves on any machine that is not. Alternatively, Franklin allows the host to compute exactly when to execute a step with each motor, and the firmware will do them at the requested time, regardless of mechanical design. However, due to limits on the bandwidth of the serial port, this ideal situation only occurs at very low speeds (below 200 steps per second, which equates to around 4 mm per second, depending on the printer design). At higher speeds, the moves are interpolated in the same way as Pacemaker and most other firmwares handle non-Cartesian configurations. Franklin Firmware handles temperature controls, general purpose input/output (GPIO) pins, and control of stepper motors.

2.3.2.1 Temperature Controls

The Arduino has an analog-to-digital converter (ADC) multiplexed to several pins. On one or more of those pins, a voltage divider with a thermistor is normally used to measure the temperature of extruders and the heated bed of conventional RepRaps. Franklin Firmware continuously reads all analog inputs that are set up as temperature controls and controls two GPIO (general-purpose input/output) pins based on the measured value. One of the controlled GPIOs is connected to the heater and the other controls a cooling fan. The heater and fan controls each have set points and the state of the GPIOs upon reaching set point is configurable. It is also possible to simply measure temperature without any control activity. The temperature reading is communicated to the host, so it can be used by scripts and shown in a browser.

Updating the heater and fan is not time sensitive so it is reasonable to take that part of the controls out of the firmware and implement it in the driver instead. However, it is important that these controls never fail: if a heater remains switched on while the extruder is very hot, it will damage the machine and can be a fire hazard. Were temperature control implemented in the Franklin Driver instead of Franklin Firmware, this scenario could occur when the serial connection is lost. Therefore, a simple heater control system is integrated into the Franklin Firmware. A more complicated system, such as a PID controller, could still be implemented in the Franklin C Driver. In that case, the system in the Firmware would serve as a protection against overheating.

2.3.2.2 GPIO Controls

Some printers and CNC machines have a need for controlling extra components. In most cases, they need simple digital signals, which can be either on or off. For example, the open source metal RepRap 3-D printer needs to switch a metal inert gas (MIG) welder (also known as gas metal arc welding or GMAW) on and off. Every microcontroller has a number of digital GPIO pins that can be used for this purpose. Franklin Firmware provides a simple interface to use them: it allows the pins to be set in any of four states: output low, output high, input, or input with pull up resistor. Scripts can read the values from input pins. In addition to storing a current state of each pin, it also stores a reset state. When a reset is requested, or when the connection is lost for some time, the pins are all changed to their reset state to avoid problems (for example, the welder will be switched off upon permanent loss of communication with the host).

2.3.2.3 Stepper Motor Control

The hardware may have any number of stepper motors to move the tool in space, and to activate the tool (such as to extrude material from the nozzle in fused filament fabrication). However, the motors do not always correspond directly to a simple direction in Cartesian space. For example, on a delta bot, all three positioning motors need to move simultaneously for the tool to move in any straight line. As the conversion from tool position to motor positions can be complex, this conversion is performed in the Franklin C Driver.

Franklin Firmware only has a concept of motor positions. The Franklin C Driver sends it a list of numbers of steps to take, which can be positive or negative. Franklin Firmware steps through this list at a constant speed and sends the steps to the motors at the requested times. ADC readings are performed in the background, leaving the microcontroller to focus resources on motor control so that steps can be very well timed leading to smoother movement.

2.3.3 Franklin C Driver

The computationally intensive conversion from tool position to motor position is implemented in C++ as Python is not fast enough. It groups motors into “spaces”. Every space has a Cartesian coordinate system where the tool can be positioned, and a certain geometry. Currently, Cartesian and delta geometries are implemented, while other geometries are left for future work. 3-D printers will normally consist of two spaces: one for moving the 3-dimensional tool and one for the 1-dimensional extruder. A 3-D printer with multiple extruders has a higher dimensionality for its extruder space.

The Franklin C Driver receives instructions similar to G-code from the Franklin Python Driver. It converts those into the step maps that the Franklin Firmware needs, and sends them to the printer controller. In addition, it passes GPIO and ADC commands between the Franklin Firmware and the Franklin Python Driver.

2.3.4 Franklin Python Driver

As many tasks as possible are implemented in the Franklin Python driver, including interface provision with both scripts and browsers. The Franklin Python driver informs interfaces of changes, parses G-code they send, and stores settings. All settings, including pin assignments, can be changed without rewriting the firmware to the microcontroller. In addition, settings can be written to hard disk as machine profiles to make sure they are available for reloading the next time the machine starts. Any number of profiles are supported permitting settings for different purposes. For example, if a machine can accommodate different tools, a different profile may be used for each tool.

2.3.5 Franklin Server

Multiple machines are handled concurrently by the Franklin Server. It detects devices when they are connected and starts the Franklin Python Driver for them, which in turn starts its own Franklin C driver. If a device is detected for which a Franklin Python Driver had already been started (that is, one that had lost its connection), the Franklin Server informs this driver of the reconnected machine and it will resume its operation. If this happens before the buffer in the Franklin Firmware runs out, it has no effect on the currently running job.

The Franklin Server provides a web interface, similar to Octoprint. (but without a webcam, although this is easily added). This means that any device with a browser, including a tablet or smartphone, can be used to control it. The web interface makes heavy use of Javascript to present all the changes that are reported by the Franklin Server to the user. It communicates with the Franklin Server using websockets. This same websockets interface is also exposed to other scripts. The web site allows manual control of the machine while scripts can integrate it with any automated or manual system.

The web interface allows all settings to be changed, including pin assignments, thermistor values and printer geometry (Figure 2.2). These changes take effect immediately. For example, if a request to double the number of steps per millimeter for the extruder drive is received while printing a segment, the remainder of that segment outputs twice as much filament per millimeter than before the change.

Spaces:	2					
Temps:	1					
Options:	1					
Spaces	Name	Type	Max Deviation (mm)			
Position		Cartesian	Set	Delta	Max Deviation 0.00	
Extruders		Cartesian	Set	Extruder	Max Deviation 0.00	
All Spaces		Cartesian	Set		Max Deviation	
Cartesian/Extruder	Number of axes					
Extruders	1					
Axes	Name	Park pos (mm)	Park order	Max v (mm/s)	Min (mm)	Max (mm)
X		0.0	1	150	-200.0	200.0
Y		125.0	1	150	-200.0	200.0
Z		160.0	0	100	0.0	160.0
All axes 0						
E0		NaN	0	100	-Infinity	Infinity
Motor Settings	Name	Home order	Limit v (mm/s)	Limit a (mm/s²)		
U		0	150	4000.0		
V		0	150	4000.0		
W		0	150	4000.0		
All motors 0						
E		0	100	10000.0		
Motor Hardware	Coupling (steps/mm)	Microsteps	Switch pos (mm)			
U	53.333	16	241.100			
V	53.333	16	241.900			
W	53.333	16	241.700			
All motors 0						
E	90.000	16	0.000			
Delta	Min Axis Distance (mm)	Max Axis Distance (mm)	Rod Length (mm)	Radius (mm)		
U	0.0	Infinity	250.243	125.100		
V	0.0	Infinity	250.243	125.100		
W	0.0	Infinity	250.243	125.100		
All motors 0						
Delta	Angle					
Position	0.00					
Extruder	Offset X	Offset Y	Offset Z			
E0	0.0	0.0	0.0			
Temp Settings	Name	Fan temp (°C)	Bed			
E		50				
Temp Hardware	R0 (kΩ) or a (1000)	R1 (kΩ) or b (1000)	Rc (kΩ)	Tc (°C)	B (1) or NaN	
E	10.0	Infinity	100.0	20	3885	
Gpio	Name	Reset state	Fan	Spindle		
Fan		Off	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
LED	D28	<input checked="" type="checkbox"/> Valid	<input type="checkbox"/> Inverted	<input type="button" value="Set"/>		
Probe	D27	<input checked="" type="checkbox"/> Valid	<input checked="" type="checkbox"/> Inverted	<input type="button" value="Set"/>		
U						
Step	D15	<input checked="" type="checkbox"/> Valid	<input checked="" type="checkbox"/> Inverted	<input type="button" value="Set"/>		

Figure 2.2: Franklin web interface.

Additionally, if a change in machine geometry is made while the motors are enabled and idle, the tool is moved to the current position according to the settings describing the new geometry. This allows for easy machine calibration, including calibration types that are challenging with traditional systems such as the printer radius of a delta. Franklin instructs the printer to move to a height of 0. Then the limit switch positions are changed and the nozzle will move to reflect that change. When the nozzle touches the build platform, the limit switch position is correct. Then the nozzle is instructed to move horizontally to the edge of the build platform. Now the radius is changed and the nozzle will again move to reflect that change. When the nozzle touches the build platform again, the calibration is complete. The whole procedure can be completed within one minute with no additional sensors.

2.3.6 Communication

There are four communication interfaces in Franklin. The first is a websockets connection to the browser or script clients. Every packet on this connection is a JSON array, containing a numerical ID chosen by the client, the name of the function to

call, a list of regular arguments and a list of keyword arguments. The server calls the requested function with the given arguments and returns either a value or an error, also encoded as a JSON array. This array contains the ID that was sent with the request, so the client can send asynchronous messages. A list of supported functions is given in Tables 2.2 and 2.3.

The second interface is between the Franklin Server and the Franklin Python Driver. The commands are sent over standard input and standard output of the Franklin Python Driver. The protocol is identical to the one described above, except that the JSON packets are not wrapped in a websocket before being sent. When the functions in Table 2 are requested from the Franklin Server, it will pass them through to the Franklin Python Driver. The exception are the functions marked “server only”; those are handled by the server and not valid for this interface.

The third interface is between the Franklin Python Driver and the Franklin C Driver, and is different from the others. The Franklin C Driver should not be burdened with parsing JSON packets, so instead it accepts binary data which is more easily extracted from the packet. Every packet starts with a byte containing the length of the packet, followed by a command byte. Each command has its own arguments. A list of all commands that the Franklin C Driver understands is given in Tables 2.4 and 2.5.

Commands from the Franklin C Driver to the Franklin Python Driver also start with a length byte and a command byte. Then follow three 32 bit integers and a 32 bit floating point value. For most commands, 18 bytes are all that is required. The commands that may need more bytes (DATA and RUN_FILE) have a larger length value and follow those values with a list of bytes.

When a GOTO or PROBE command is received, a reply is sent, which is either OK or WAIT. In the latter case the queue is full and no further GOTO or PROBE commands can be sent until the Franklin C Driver sends a CONTINUE command.

The last interface is between the Franklin C Driver and the Franklin Firmware. This interface has unique qualities because it is implemented over a physical serial line that is much slower and unreliable. Because of the large amount of data that needs to be transmitted over this slow interface, the length byte is not sent, and the length of packets must be determined from their content. Because reliability is important, a checksum is added to maintain packet integrity.

Data is either a single byte command or a packet. The single byte commands all have their highest bit set, and are otherwise chosen to have a maximum distance between

Table 2.2
Supported functions.

Command	Function
upload	Flash new firmware to an Arduino. (server only)
find_printer	Search for a machine by UUID or port. (server only)
set_autodetect, get_autodetect	Whether the server tries to detect a machine on newly discovered ports. (server only)
disable	Deactivate the machine. (server only)
detect, detect_all	Detect a machine, or all printers. (server only)
add_port, remove_port	Notify server of port discovery. (server only, called from kernel signals)
get_ports	Get list of available ports. (server only)
set_default_printer, get_default_printer	Which machine is used by new connections. (server only)
set_printer, get_printer	Which machine is used by this connection. (server only)
set_monitor, get_monitor	Whether this connection should be informed of changes in settings. (server only)
reset	Reset the Arduino.
die	Close the Python and C driver.
flush	Wait for queue to be empty.
probe	Map an area using a probe signal or manual feedback.
goto	Move motors to a position.
gotocb	Move motors to a position and wait for it to arrive.
sleep	Change sleep state of the motors.
settemp	Change the set point of a temperature control.
waittemp	Signal an alarm when a temperature enters a specified range.
readtemp	Request current temperature.
readpin	Request current value of a GPIO pin.
load, save, list_profiles, remove_profile, set_default_profile	Profile management.
abort	Disable heaters, sleep motors, reset GPIO pins and stop any running G-code.
pause	Pause or resume the currently running G-code.
queued	Request length of the queue.

Table 2.3
Supported functions. (Continued)

Command	Function
home	Recalibrate the machine with limit switches.
park	Home if required, then go to park position.
wait_for_temp	Wait until an alarm from waittemp is triggered.
clear_alarm	Clear the waittemp alarms.
export_settings	Retrieve all settings for storing in a text file.
import_settings	Change settings from a text file.
gcode_run	Run (parsed) G-code.
request_confirmation	Wait for the user to press a button or abort.
confirm	Signal confirmation.
queue_add	Parse G-code and add it to the queue.
queue_remove	Remove an entry from the queue.
gcode_parse	Parse G-code and return the result.
gcode_bbox	Find the bounding box of parsed G-code.
queue_print	Send one or more queue entries to the machine.
queue_probe	Probe the combined bounding box of one or more entries, then send them to the machine.
get_globals, set_globals	Manage global settings
get_axis_pos	Get current position of an axis.
set_axis_pos	Set current position of an axis, without moving the motors.
get_space, get_axis, get_motor, set_space, set_axis, set_motor	Manage space settings.
get_temp, set_temp	Manage temp settings.
get_gpio, set_gpio	Manage GPIO settings.
send_printer	Request current state of a machine.

them in terms of bit flips required to go from one to the other. A packet starts with a command, none of which has the highest bit set. The command byte is followed by arguments. One or more checksum bytes are added at the end of the packet.

In G-code, the checksum is computed by summing all bytes of the packet and using the lowest 8 bits of the result. This is very weak, and two flipped bits have a large chance of resulting in a valid checksum, even though the packet is incorrect.

Franklin uses a Hamming code [25] with one parity byte for every three bytes of data. That byte contains five parity bits, each of which set the parity of a selected group of bits to be even. The groups are carefully chosen to maximize the distance between valid packets. Two bit flips can never result in another valid packet, and more random

Table 2.4
Commands for the Franklin C Driver.

Command	Function	Response
RESET	Reset the machine.	
GET_UUID	Get universally unique identifier for the machine.	UUID
GOTO	Add segment to move queue.	MOVECB
RUN_FILE	Run a parsed G-Code file from disk.	UPDATE_TEMP, UPDATE_PIN, CONFIRM, FILE_DONE
PROBE	Like goto, and monitor probe pin to abort move and notify Python Driver about position.	MOVECB, LIMIT
SLEEP	Enable or disable the motors.	
SETTEMP	Set a temperature target.	
WAITTEMP	Set an alarm.	TEMPCB
READTEMP	Read current temperature.	TEMP
SETPOS	Set current axis position.	
GETPOS	Get current axis position.	POS
READ_GLOBALS	Get global settings.	DATA
WRITE_GLOBALS	Set global settings.	
READ_SPACE_INFO, READ_SPACE_AXIS, READ_SPACE_MOTOR	Get space settings.	DATA
WRITE_SPACE_INFO, WRITE_SPACE_AXIS, WRITE_SPACE_MOTOR	Set space settings.	
READ_TEMP	Get temp settings.	DATA
WRITE_TEMP	Set temp settings.	
READ_GPIO	Get GPIO settings.	DATA
WRITE_GPIO	Set GPIO settings.	
QUEUED	Request queue length and optionally abort move.	QUEUE
READPIN	Get GPIO pin state.	PIN
HOME	Move away from limit switches until it no longer hits them.	HOMED

Table 2.5
Commands for the Franklin C Driver. (Continued)

Command	Function	Response
RECONNECT	Machine has reconnected at this port.	
RESUME	Resume running a file that was previously paused.	
GETTIME	Get time estimates about the current job.	TIME
(asynchronous)	Notify that the state of an input GPIO has changed.	PINCHANGE
(asynchronous)	A limit switch was triggered during a move.	LIMIT
(asynchronous)	The machine was disabled due to a timeout.	TIMEOUT
(asynchronous)	The connection to the machine was lost.	DISCONNECT

flips are very unlikely to do so.

Detection of corrupt packets is required but is not sufficient for a reliable connection. Corrupt and lost packets must also be properly handled. When it is detected that a packet did not arrive, it is sent again. This means that if the packet was received but the acknowledgment was not, duplicate packets may be received. This must also be handled. The method for this has been copied from the USB standard: Each packet has one bit which indicates if it is an even or odd packet. Even packets are acknowledged with ACK0, odd packets with ACK1. If an even packet is received after an even packet, then the original ACK must have been lost. In that case, another ACK0 is sent, but the packet is ignored, because the original even packet has already been handled. Odd packets are handled similarly.

This system allows any amount of lost or corrupted packets with no effect on the reliability of the link. Because the Franklin Server is capable of detecting that a new printer is the same as one that has previously disconnected, it can pick up the connection and continue as if nothing happened. If this happens in the middle of a print, it will only pause for a moment. If the connection is restored before the queue in Franklin firmware is empty, no pause will occur. Table 2.6 lists all the commands that are supported.

Table 2.6
Supported Commands, Function and Response.

Command	Function	Response
BEGIN	Handshake; send version and receive capabilities.	READY
PING	Handshake.	PONG
RESET	Reset the Arduino.	
SETUP	Set up globals.	
CONTROL	Manage GPIO pin states.	
MSETUP	Set up motor settings.	
ASETUP	Set up ADC (temp) settings.	
HOME	Move away from limit switches.	HOMED
START_MOVE	Begin sending movement buffers.	
START_PROBE	Begin sending movement buffers for probing.	
MOVE	Send movement buffer for one motor.	
START	Begin moving.	
STOP	Stop moving and discard buffer.	STOPPED
ABORT	Stop moving, disable all motors, reset all heaters and GPIO.	STOPPED
DISCARD	Discard a part of the queued buffers without stopping the current move.	
GETPIN	Read current state of a GPIO pin.	PIN
(asynchronous)	Buffer has been completed.	DONE
(asynchronous)	Buffer has been completed and no next buffer is available.	UNDERRUN
(asynchronous)	ADC has been measured.	ADC
(asynchronous)	Limit switch has been triggered.	LIMIT
(asynchronous)	Machine has been deactivated due to timeout.	TIMEOUT
(asynchronous)	The value of an input pin has changed.	PINCHANGE

2.3.7 Controlling Movements

A movement request to the C Driver consists of two speeds, F0 and F1, and a target position for each dimension in each space. F0 is the requested starting speed, while F1 is the requested finishing speed. The tool must move on a linear path and accelerate at a constant rate during the segment.

F0 and F1 are limited to maximum values which are set for each axis. Because of this, the common way to move fast is to request a speed of infinity. That will make

the tool move at the maximum allowed speed configured in settings.

While moving, the speed and acceleration of every motor is limited as well. On a Cartesian system, the relationship between motor speed and axis speed is linear and simple. But this is not true for other machine geometries. For example, on a delta system motors are changing speed constantly to keep the nozzle moving in a horizontal line at a constant speed. Franklin Firmware limits the motor speed and acceleration so it is able to avoid missing steps even when the tool is at the edge of the build volume where one motor must move a large distance to effect a small amount of distance by the tool. Franklin Firmware accomplishes this without slowing the system down when it is more near the center.

Setting limits on acceleration has a very negative influence on print speed, especially if there are many short segments, such as in curves with small radii. This is due to a discontinuity in the direction of the path that has an infinitely large acceleration for any speed other than zero. To solve this, Franklin Firmware has a setting for how much to deviate from the requested path. Franklin Firmware will cut the corners by that amount, and it will use this curve to gradually change the speed of all the motors.

Figure 2.3 schematically shows how the move comprising a segment is prepared. This occurs when the previous move has completely finished; in the case of a deviation from the path, the move has already started at this point (Figure 2.3). For that reason, every axis stores the distance to move for the current segment, and for the next segment. After filling those values, the speeds F0 and F1 are limited to what the axes are set to allow. Then the position where the curved connection should start is computed and finally all variables are filled with their values.

2.3.8 Probing

For 3-D printing and most other applications, the machine can be calibrated once with the assumption that the surface is flat and horizontal and it does not require recalibration. However, for milling PCB circuits, it is very important that the distance into the surface is tightly controlled. For this purpose, Franklin supports probing the surface before running a job, and using the measured probe map to correct for deviations from the horizontal flat ideal. This can also be used for example to print on top of complex geometries or to repair products.

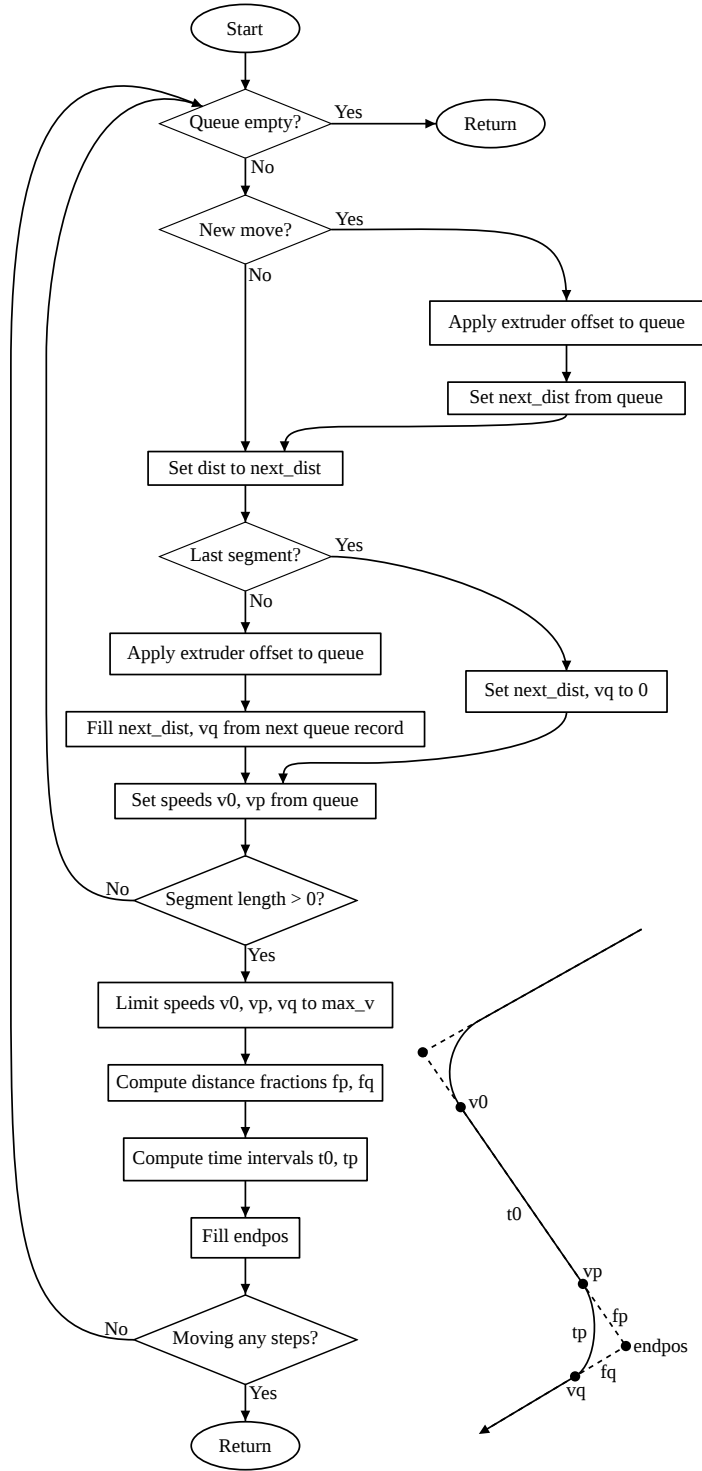


Figure 2.3: Schematic representation of how the move comprising a segment is prepared in Franklin.

2.3.9 Scripting

The above describes how Franklin is useful for controlling a standalone 3-D printer or other CNC machine. However, In a scientific environment a machine is normally part of a larger setup, rather than only a standalone 3-D printer or CNC machine. Franklin allows tight integration with the rest of the setup by supporting an extension to G-code, which allows running system commands. This can be used to control other parts of the setup, for example to record an image with a camera. Because running system commands is a security risk, this feature is disabled by default and it cannot be enabled from scripts or browsers, only at startup using a configuration file or commandline switch.

G-code does not support input other than waiting for a button. Instead, the websockets interface allows direct control over the printer without using G-code. Using this interface, a script can generate a movement pattern in real time. For example, when a camera is connected as a tool, a script could use the images from it and continuously move it to keep a moving specimen centered in the field of view and in focus.

2.4 Quality Control

All compiler warnings are enabled for both Franklin Firmware and the Franklin C Driver. Stack protection was disabled because the AVR platform does not support it. Valgrind [26] was used to find buffer overflows and use of uninitialized variables in the Franklin C Driver.

Franklin has been used extensively on a variety of devices, ranging from a 1-dimensional syringe pump to 3-dimensional printers. Firebug [27] was used to find errors in the HTML and Javascript. Franklin Server delivers web pages written in HTML 5 and Javascript (Ecma 262). Care has been taken to follow the standards; no browser extensions have been used.

Franklin has been used successfully on a variety of different machines listed above. In addition, Franklin has been used in conjunction with a metal inert gas (MIG) weld-based 3-D printer to develop substrate release mechanisms for 3-D printed parts [24]. Traditionally, metal 3-D printed parts must be removed from a print substrate with the use of saws or other machining equipment. This removal step is undesirable as it results in excessive material waste and has additional associated time and cost

requirements. In order to evaluate possible substrate release mechanisms, Franklin was used to print aluminum lap shear test specimens on aluminum and steel print substrates. This study observed that low-cost options, such as boron nitride coatings, and no-cost options, such as printing aluminum on steel substrates, minimized the amount of force required to remove metal specimens from a print substrate.

(2) Availability

2.5 Operating System

Franklin has been designed for and tested on Debian GNU/Linux [28], on i386, amd64 (PC) and armhf (Raspberry Pi [29] and Beaglebone Black [30]) architectures. It should work on any other GNU/Linux system and possibly other Unix-based systems. The packages have been built on the latest stable (jessie, or Debian 8) and unstable versions, and using backports [31] with some packages from jessie, also on oldstable (wheezy, or Debian 7).

2.6 Additional System Requirements

Franklin is only useful if there is hardware connected to control. It does not have any other requirements.

2.7 Dependencies

† Python-fhs [32]: module for reading and writing files in the proper place according to the Filesystem Hierarchy Standard [33].

† Python-network [34]: module for using network connections, including SSL.

† Python-websocketd [35]: module for hosting a web server that can communicate using websockets.

2.8 List of Contributors

Bas Wijnen (code development).

2.9 Software Location

2.9.1 Archive

Name: Purl.org
Persistent identifier: <http://purl.org/NET/mtu-most/franklin>
License: GNU Affero General Public License, version 3 or later.
Publisher: Michigan Technological University.
Date published: 09/06/2013.

2.9.2 Code Repository

Name: GitHub
Identifier: <https://github.com/mtu-most/franklin>
License: GNU Affero General Public License, version 3 or later.
Date published: 09/06/2013.

2.9.3 Language

All code, comments and documentation are in American English.

(3) Reuse Potential

In its current state, Franklin can be used for controlling 3-D printers and other manufacturing machines to produce research-grade equipment. Such equipment can also be controlled with Franklin.

While the web interface is targeted at 3-D printers and similar machines, the code is written in a way that makes it easy to design an alternative interface. For example, a syringe pump could be controlled with an interface that allows programming a sequence of flows in the way that users of syringe pumps expect. Such a program would use the websockets interface of Franklin to control the hardware, while presenting the user with an interface that is more appropriate for the application than the default CNC interface.

Because it is free software users can improve Franklin and the hardware to fit their requirements. Franklin is in active development and the developer can be contacted through github.

Possible improvements, which are considered or being worked on include:

- † G-code parsing is presently time consuming and processor intensive. Performance can be improved through the use of a compiled language such as C++ instead of Python.
- † One-click printing support like that common for 2-D printers would vastly improve ease-of-use. This functionality is a logical extension to common 3-D modeling packages (Blender, OpenSCAD, FreeCAD, etc.) and even web browsers e.g. when browsing objects hosted on 3-D printer aggregate sites such as Youmagine.com, 3dprint.nih.gov or Thingiverse.com.
- † Running a G-code converter (a slicer for 3-D models) as part of Franklin would allow users to send model files directly to Franklin, again improving ease-of-use. (This is a requirement for the previous point, but is also a useful feature in itself.)
- † Handling a microcontroller reset or loss of power on the entire system, including the host computer would be useful for those using 3-D printers in the developing world, where power is less reliable. It would also be useful when using very large 3-D printers, such as the Gigabot, which print for many hours to produce a single object.

- † Because Franklin parses G-code before running it, and because it does this on a relatively powerful computer, it has the option to thoroughly analyze it and take action based on the relatively distant future.
- † The BeagleBone contains all the features that are needed for the microcontroller and the host. It would be possible to use just the BeagleBone for the entire control system.

2.10 Conclusions

Firmware known as Franklin was developed to mitigate limitations associated with other CNC and 3-D printer control systems. Franklin was successfully demonstrated on a wide range of RepRap-derived devices: Mendel RepRap, Delta RepRap, Quad Delta RepRap, OS laser welder, PCB micromill and the open source metal 3-D printer. Franklin demonstrated the ability to support the maker movement with low-cost open-source control of three dimensional additive and subtractive fabrication as well as scientific analytical equipment. Low-cost RepRap 3-D printers are being taken more seriously in the scientific and industrial worlds. Franklin improves upon this by allowing more functionality and better integration with new or existing systems.

2.11 Competing Interests

The authors declare that they have no competing interests.

2.12 Acknowledgements

The authors would like to acknowledge valuable discussions with Lars Pötter, and Markus Hitter on the RepRap mailing list, reprap-dev@lists.reprap.org.

Chapter 3

Open-Source Mobile Water Quality Testing Platform

Abstract¹

The developing world remains plagued by lack of access to safe drinking water. Although many low-cost methods have been developed to treat contaminated water, low-cost methods for water-quality testing are necessary to determine if these appropriate technologies are needed, effective, and reliable. This paper provides a methodology for the design, development, and technical validation of a low-cost, open-source (OS) water testing platform. A case study is presented where the platform is developed to provide both the colorimetry for biochemical oxygen demand/chemical oxygen demand and nephelometry to measure turbidity using method ISO 7027. This approach resulted in equipment that is as accurate, but costs between 7.5 and 15 times less than current commercially available tools. It is concluded that OS hardware development is a promising solution for the equipment necessary to perform water quality measurements in both developed and developing regions.

¹The material contained in this chapter was previously published in the Journal of Water, Sanitation and Hygiene for Development[36]

Water quality is a major problem in the developing world as roughly 780 million people are still without safe drinking water [37]. Every year about 760,000 children die from diarrhea [38] – largely caused by a lack of clean drinking water and sanitation [37, 39, 40]. There are many low-cost methods to provide drinking water with safe levels of known biological and chemical contaminants [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. For example, solar water disinfection (SODIS) has been proven in both bench and field scale trials to significantly reduce microbial content in contaminated water, and associated incidence of diarrhea in users, and techniques have been developed to make SODIS usable in most places in the world [51, 52, 53, 54, 55, 56, 57, 58]. While these simple water treatment methods have demonstrated efficacy, there has been little development of simple and inexpensive water quality assessment instrumentation. This sort of instrumentation is required to know whether or not such methods are: (1) needed, (2) effective, and (3) reliable (e.g. in the case of filter blinding). They must have a minimal cost in order to be deployed in the field in developing regions.

One promising method to obtain high-quality scientific tools while radically reducing costs is to use an open-source (OS) hardware approach [10, 59]. Due to the tremendous success of free and open source software development [60], the concept of open source has spread to areas of both appropriate technology for sustainable development [61, 62, 63] and other hardware [64] such as 3-D printers [2, 3, 20], which in turn can be used to fabricate OS scientific tools [10, 59, 65]. This approach was used recently to combine the OS Arduino electronics prototyping platform and the RepRap 3-D printer to make an OS colorimeter, which could be used for water testing using the chemical oxygen demand (COD) method [66]. This open-source hardware (OSH) approach is expanded here to create a platform which could be used for a collection of water tests.

This paper provides a methodology for the design, development, and technical validation of an OS water testing platform. A case study is presented where the platform is developed to provide both the colorimetry for biochemical oxygen demand (BOD)/COD and nephelometry to measure turbidity using method ISO 7027 [67]. This approach is evaluated for its potential to reduce the cost of equipment to perform these measurements of water quality and the results are discussed to provide conclusions about the future of water testing in developing regions.

3.1 Methods

3.1.1 Design Methodology

The schematic for the as-built OS water testing platform that was designed to perform colorimetric COD/BOD and nephelometry is shown in Figure 3.1.

The OS water testing platform case design was wholly completed in OpenSCAD 2013.06.09, a freely available, OS, script-based solid modeling software. The assembled case with electronics is shown in Figure 3.2 and the design of the case body is shown schematically in the inset. The case was printed with a RepRap 3-D printer with black poly-lactic acid media so as to minimize stray light inside the detection area.

The electronics are based upon the OS Arduino prototyping platform, which is designed to use ‘shields’ or customized electronic boards that can be pressed into place and that typically come with software libraries so as to facilitate integration of board features into the custom code developed by the end user. As can be seen in Figure 3.1, multiple LEDs and light intensity sensors are connected directly to the Arduino’s digital inputs and outputs. The microcontroller contains flash memory to store a program for performing the measurement and providing a user interface using a shield containing a character LCD screen and navigation/control buttons.

A total of three discrete electronic components are required for the nephelometer circuit (in addition to the Arduino and the shield); an additional two components are necessary for the colorimeter functionality, as shown schematically in Figure 3.1. The device’s firmware (<https://github.com/mtu-most/colorimeter>) provides an easy to navigate hierarchical menu system for selection of device functions. The firmware can be changed and rewritten to the device using the Arduino IDE, which is distributed as free and open source software from the Arduino website (<http://www.arduino.cc/>).

There are multiple standards for measuring turbidity [68]. The USEPA method 180.1 (EPA, 1979) was the first standard promulgated, but it suffers from poor reproducibility [68, 69]. A newer standard is maintained by ISO [67], which tries to avoid some of the problems of the EPA method. It requires an infrared photodiode instead of a tungsten lamp, which improves measurement of turbidity resulting from the presence of biological material, which may not be measured by the USEPA method. The ISO

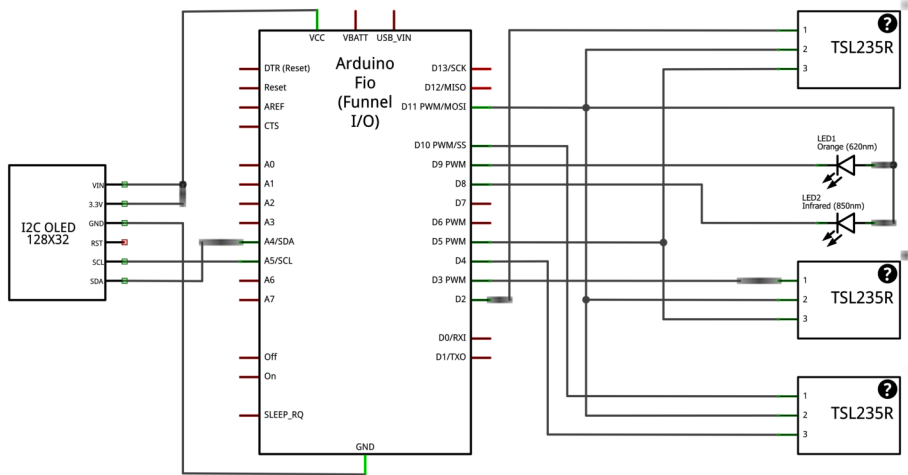


Figure 3.1: Schematic of the OS mobile water quality testing platform.

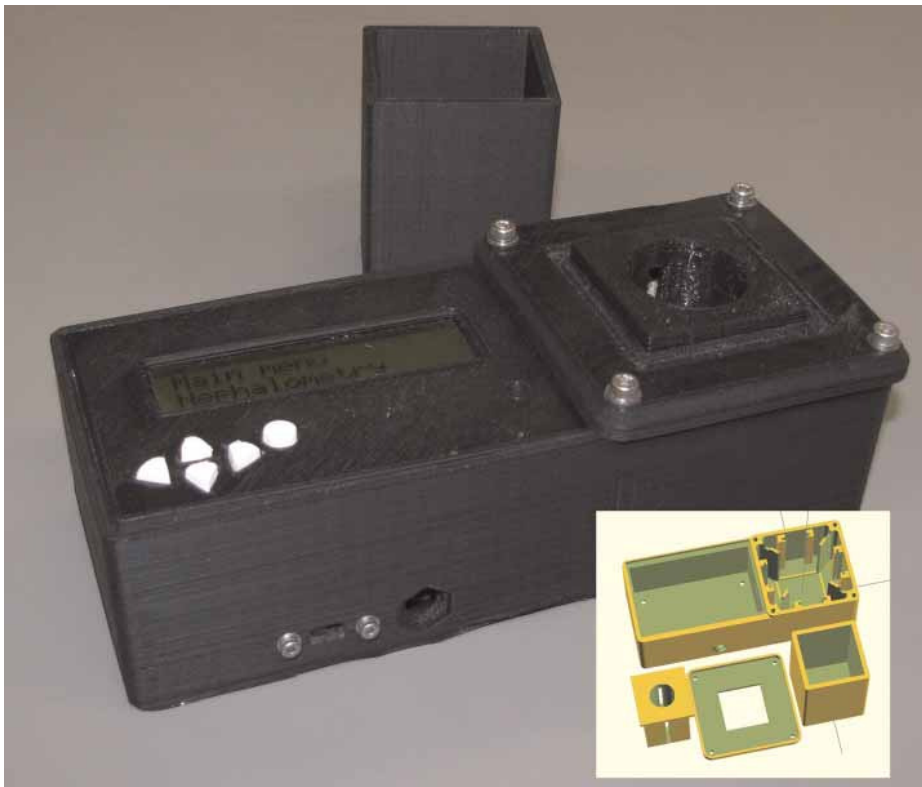


Figure 3.2: OS water testing platform with the assembled case, electronics and an inset schematic of case design in OpenSCAD.

method was chosen for this design as it has demonstrated greater reproducibility, utilizes a low-cost, long-life and less energy intensive diode light source and has greater sensitivity when biological material is present in the sample.

The light intensity sensor used is a TAOS TSL235R light-to-frequency converter. This device produces a digital signal with a frequency proportional to the amount of light that it detects. The Arduino's internal counter units are connected to it, permitting measurements up to 8 MHz. Both of the Arduino's two counter input pins are connected to different sensors. During measurement, one sensor measures the intensity of the LED directly (reference intensity) and a second sensor measures either transmitted or diffuse reflected light. Measurement of the reference (LED intensity) and the transmitted/reflected light intensities are made simultaneously. The reference measurement is used to compensate for any changes in illumination intensity, which can occur due to fluctuations in supply voltage, temperature or aging of the diode.

Because this device is similar to the OS colorimeter [66], both can be combined into one device. To include colorimetric capability, only one additional LED and light intensity sensor are required, the cost of which is negligible.

3.1.2 Performance measurement methodology

The performance of the COD colorimeter was reported earlier [66] and so was not investigated as part of this research.

Turbidity standards (Hach StablCal Turbidity Standards Calibration Kit) were used to calibrate both the OS nephelometer and a Hach 2100P portable turbidimeter, an instrument in the same category as the OS nephelometer. Formazin Turbidity Standard (4,000 NTU) was diluted with distilled water to produce samples having turbidities of 2, 4, 40, 100 and 200 NTU and each of these was analyzed in triplicate with the two instruments.

3.2 Results and Discussion

The results from calibration and a least squares fit are shown in Figure 3.3. Measurements from the two calibrated instruments are compared and are shown in Figure 3.4 with the line indicating perfect agreement between the two instruments (line of unity).

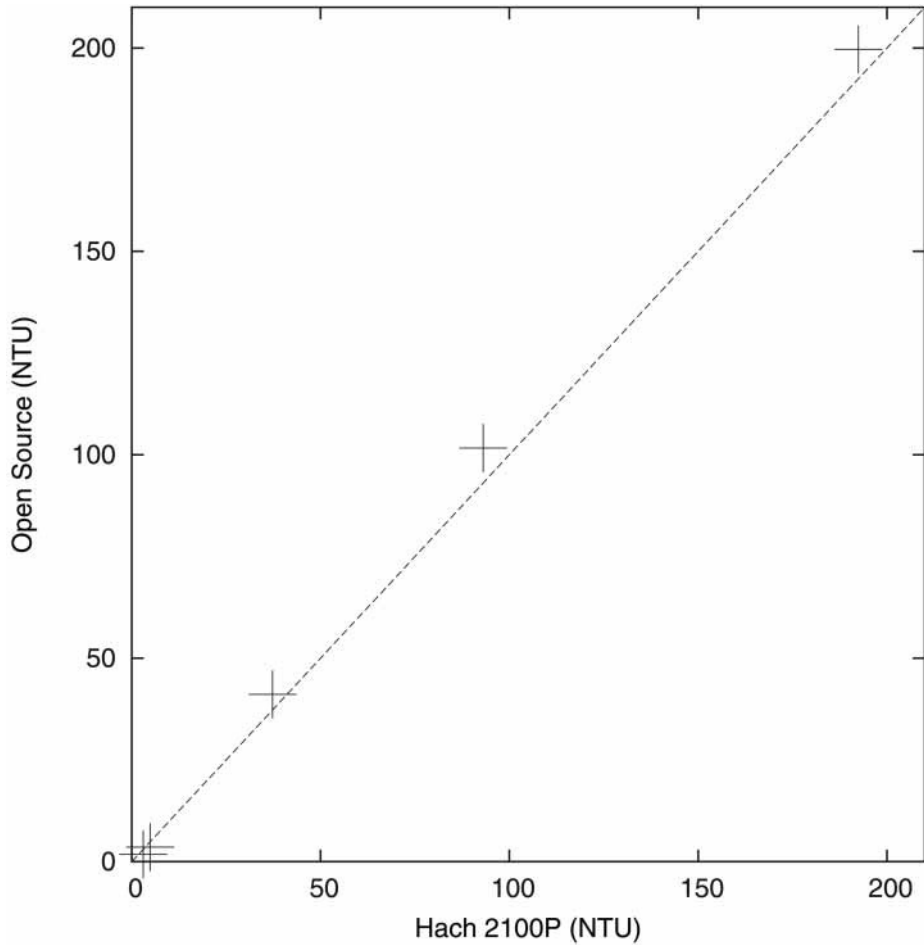


Figure 3.3: Calibration of the OS nephelometer. The line shown is the result of the calibration.

A device for measuring turbidity is presented, which in combination with other tests, is useful for determining whether water is safe for drinking. It has a direct application in determining the exposure times necessary for safe drinking water for those using the SODIS method or UV disinfection. This is especially a concern in developing countries, where people do not have money to buy the relatively expensive, commercially available nephelometers, which range in price from \$600 to \$1,200, for equivalent quality to what is demonstrated here. The OS device performs as well as the commercial version it was compared to in this study, and for under \$80.00, costs 7.5 to 15 times less. Low-cost and localized fabrication and maintenance of the device make it feasible for communities to do scientific measurements at both distributed and centralized water treatment facilities without depending on external support.

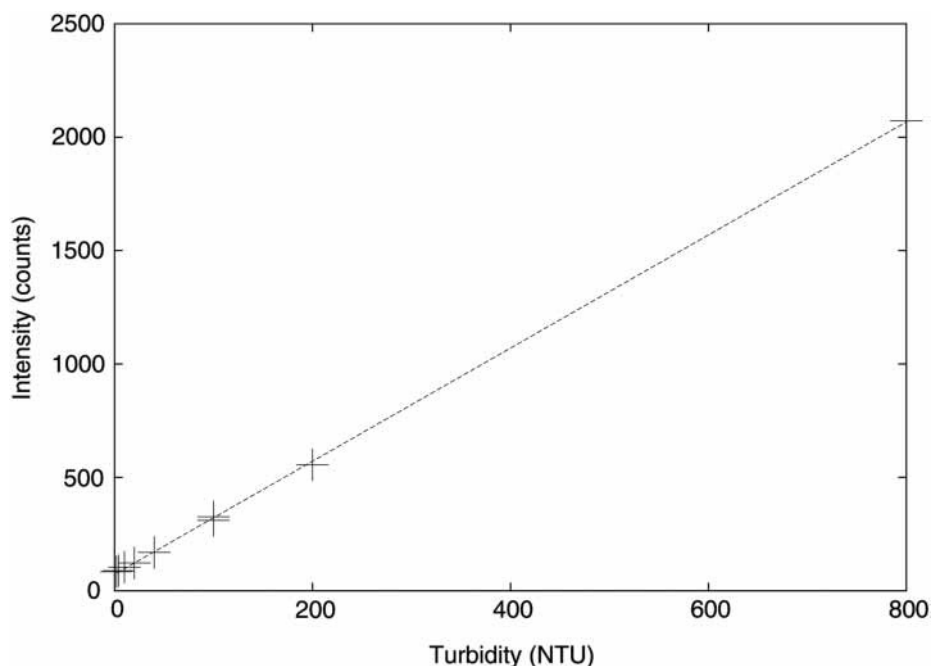


Figure 3.4: Comparison of the OS nephelometer and the Hach 2100P. Points which have identical readings on both devices are on the line.

3.3 Limitations and Future Work

The drawbacks to an OSH approach to scientific tools are outlined in the ‘Open Source Lab’ [59] and found to be without merit, particularly concerning the lack of incentives for innovation. The one aspect of OSH that is very important for scientists is accuracy, which has been demonstrated for this particular device with the results of this study. For any other OSH for science a similar study is necessary.

It should be pointed out that BOD/COD and turbidity are not enough to ensure safe water alone. The device brings the price of a nephelometer down by an order of magnitude. With the simple addition of more sensors, at minimal incremental cost, it replaces more devices designed to perform other water quality test methods, breaking the profit-by-specialization paradigm that drives commercial instrument development. In this way, a complete water testing lab may be built in pocket format in the future, containing not only turbidity and COD, but also pH, solid dissolved substances, temperature, total dissolved solids, total suspended solids, dissolved oxygen, oxidation reduction potential, ultraviolet transmittance and BOD.

Because the hardware and software are all OS, this is an ideal platform for adding more sensors or actuators to increase functionality. The software for controlling the new part can simply be added to the existing code base and it will work. This means any specialized measurement device can be built with relatively little effort. For instance, the device as tested does not contain the extra sensor for doing a ratio measurement shown to improve sensitivity by accounting for absorption [69], but this sensor can easily be added because the software is open source; the user who is interested in this feature can implement it without help from an external manufacturer.

Because the design of the platform developed here is so flexible, many extensions are possible to make it more useful. For example, to allow the device to be used in places without access to a power grid for long periods of time, it can be charged with a small, solar photovoltaic cell or hand crank. To make it more flexible to use, it could be helpful to add wireless networking support, which would also reduce the cost as it would allow removal of the display and buttons and instead use the ubiquitous smart phone to control it. The phone could then be used to connect to several devices, and integrate several parts of an experiment, or do multiple separate experiments simultaneously. Using the power of 3-D printing the device could be configured as a phone attachment or could be used in conjunction with the OS tricorder project [59]. Future work is also needed to test the platform in the field outside of controlled laboratory conditions. In addition, future work is needed to gauge the social acceptability of the device for any intended application, real world acceptance and if necessary marketing. To determine if the tests are effective, both field and clinical trials may also be useful. Because the software is open source, the communication protocol is automatically public as well. As an addition to that, it would be useful to design an open standard, which can be used by controlling computers to communicate with any such measurement device, without even knowing what type of measurement it is doing. Due to the low cost, this device, or a variation of it, would be very suitable for educational settings, which often function under severe budget constraints.

3.4 Conclusions

The results of this study have shown that a valuable device for testing water can be built using OSH and software, for a fraction of the price of commercial options. Such a device not only matches the performance of commercial devices, but it is also expandable to enable other testing procedures at minimal additional cost. This method of developing and fabricating scientific testing equipment is valuable to all scientists, but may be particularly attractive to anyone in need of water-quality testing who, for financial reasons, would not have access to the instrumentation necessary.

3.5 Acknowledgements

The authors would like to acknowledge helpful discussions and input with S. Feeley, D. Perram and crowd-sourced funding from supporters through Superior Ideas.

Chapter 4

Open-source Syringe Pump Library

Abstract¹

This article explores a new open-source method for developing and manufacturing high-quality scientific equipment suitable for use in virtually any laboratory. A syringe pump was designed using freely available open-source computer aided design (CAD) software and manufactured using an open-source RepRap 3-D printer and readily available parts. The design, bill of materials and assembly instructions are globally available to anyone wishing to use them. Details are provided covering the use of the CAD software and the RepRap 3-D printer. The use of an open-source Raspberry Pi computer as a wireless control device is also illustrated. Performance of the syringe pump was assessed and the methods used for assessment are detailed. The cost of the entire system, including the controller and web-based control interface, is on the order of 5% or less than one would expect to pay for a commercial syringe pump having similar performance. The design should suit the needs of a given research activity requiring a syringe pump including carefully controlled dosing of reagents, pharmaceuticals, and delivery of viscous 3-D printer media among other applications.

¹The material contained in this chapter was previously published in PLOS One [13].

4.1 Introduction

Free and open source (or libre) technological development is a fundamentally new, decentralized, participatory and transparent system to create both software and hardware. It stands in sharp contrast to the closed box, top-down, and secretive standard commercial approach to development [60]. As much of the Internet now relies on free and open-source software (FOSS), open source is becoming the norm in software development [70, 71]. FOSS has been so successful that for many applications it is the defacto standard, with 94% of the World's top 500 supercomputers, 75% of the top 10,000 websites and 98% of enterprises using open-source software [72, 73]. FOSS is computer software made available as source code (open source) that can be used, studied, copied, modified, and redistributed without restriction, or with restrictions that only ensure that further recipients have the same rights under which it was obtained [74]. FOSS is in widespread use in science and engineering and has driven down the cost of numerical simulation in a number of fields ranging from psychotherapy [75] and medicine [76, 77], neural circuit reconstruction [78], genomic sequences annotation [79], education [80, 81], and ecology [62]. In addition, it has been proposed as a solution to the intellectual property tragedy in nanotechnology, which has slowed progress and deployment in the field [11, 82, 83, 84]. Even greater cost reductions for science, however, can be found with the application of open source hardware [10, 59, 65, 66]. The development of open-source hardware has the potential to radically reduce the cost of performing experimental science and put high-quality scientific tools in the hands of everyone from the most prestigious labs to rural clinics in the developing world [10, 16, 59].

This article introduces a low-cost open-source family of syringe pumps. Creation of parametric open-source designs using an open-source computer aided design (CAD) package is described to produce customized syringe pumps for scientific and/or health applications. Details are provided for use of open-source RepRap 3-D printers to fabricate the components. An open-source Raspberry Pi computer used as a wireless control device is also illustrated. The performance of the pumps produced is assessed and the method's advantages, known limitations and potential for radically reducing the cost of doing science are discussed.

4.2 Materials and Methods

The low-cost open-source family of syringe pumps are completely customizable allowing both the volume and the motor to scale for specific applications. The bill of materials for the three variations of the syringe pump are shown in Table 4.1². The user/designer must first determine the size of motor to enable that application. The appropriate motor size can be selected once the required torque is known following [85]. A bigger motor provides more torque, but necessitates larger printed components. A bigger syringe allows more fluid to be pushed out, both per second and in total, but decreases the precision of the device. A simple change to the OpenSCAD script specifying the motor selection defines the dimensions for the printed parts.

4.3 OpenSCAD and Open-source 3-D Printing

Open-source and freely available OpenSCAD is script-based, parametric CAD software possessing powerful 3-D modeling capabilities [86]. It is not graphical; models are created by adding and subtracting primitives to produce the desired shape. It supports creation and extrusion of polygons and poly lines, so can be used to create very complex shapes. The script language is based upon C++ and only a few methods are required to produce very complex designs, so the learning curve is short, albeit steep for those not possessing programming experience. The scripts are written such that designs are parametric — the design can easily be altered by changing key dimensions. For instance, the syringe pump script can be altered to produce parts fitting different motors simply by specifying which motor to design for. The script written for the syringe pump is available online [87]. Models rendered in OpenSCAD are typically exported as stereolithography (stl) files for the first step in producing a 3-D print using any of the RepRap 3-D printers currently available. Images of syringe pump parts rendered by OpenSCAD and photographs of the printed parts are shown in Figure 4.1.

RepRap printers almost universally require g-code, a human-readable file format specifying the path the print head must follow to produce a physical object from a software model. G-code is produced by software referred to as a “slicer”, which, as the name implies, slices an stl model into layers each having the same thickness in the z-direction. Cura was used to slice the syringe pump stl models [88]. Cura is also open-source and freely available.

²The Raspberry Pi has decreased significantly in price since the publication of this paper.

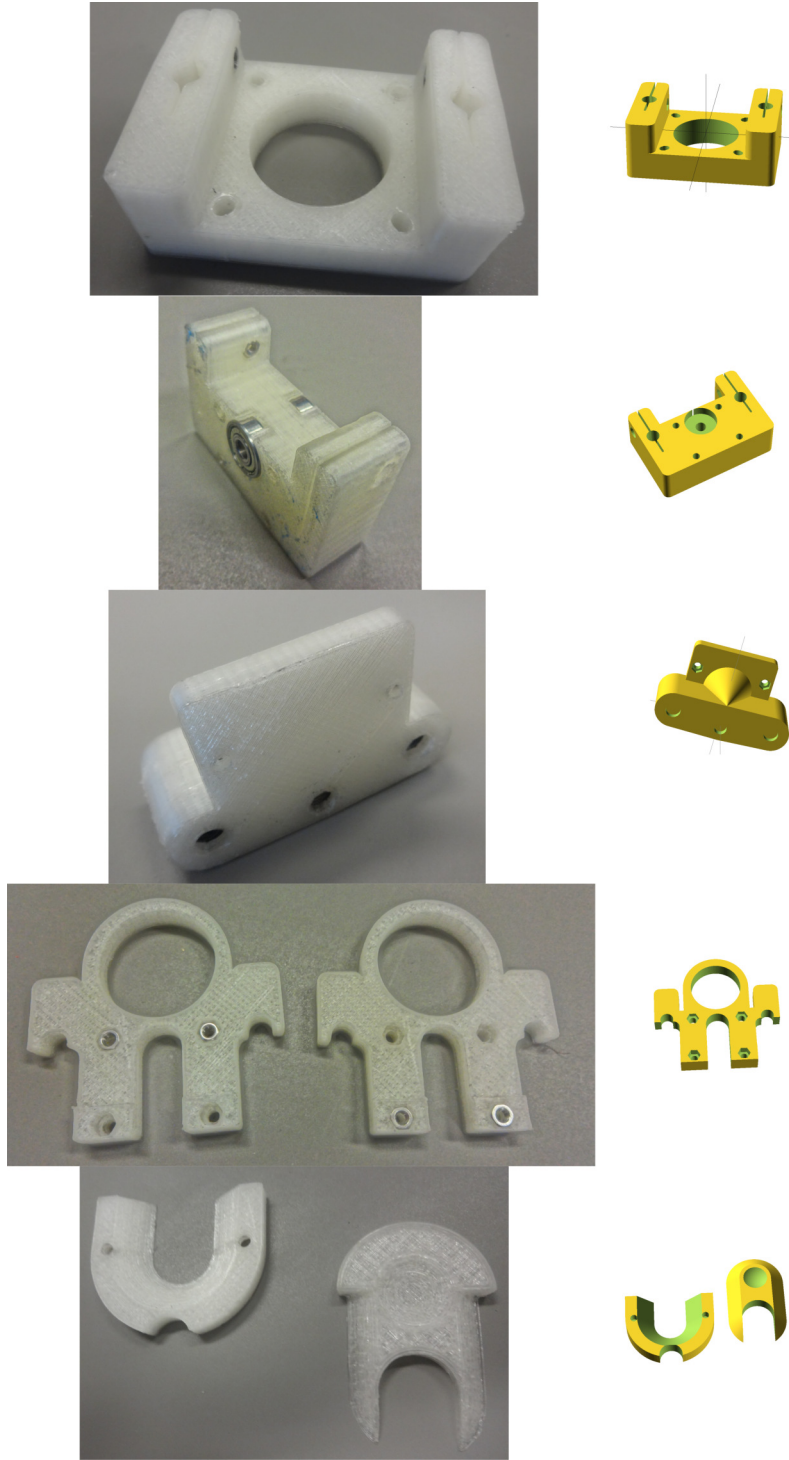


Figure 4.1: 3-D printable parts for the open-source syringe pump.

Table 4.1

Bill of Materials for three examples of the open-source syringe pumps. All parts are printed in PLA with 20% infill. The cost of a dual NEMA17 pump is twice that of a single NEMA17 pump, except only one Raspberry Pi is needed.

Part	Cost (US\$)
Printed Motor End	1.33/1.91
Printed Idler End	1.40/1.88
Printed Carriage	1.59/1.58
Printed Syringe Clamp	1.35/3.27
Printed Plunger Wedge	0.60
NEMA11/NEMA17 motor	15.95/19.95
M3x40 bolt (4x)	0.35
M3x16 bolt (4x)	0.51
M3x12 bolt (6x)	0.64
M3 washer (8x)	0.14
M3 nut (10x)	0.10
M5 nut (5x)	0.09
M5 threaded rod (1 m)	5.03
6 mm smooth steel rod (1 m)	11.03
lm6uu Linear bearing (2x)	4.66
635zz Ball bearing (2x)	1.50
Coupler	3.49
Raspberry Pi	40.00
Total	89.76/96.72

The parts were printed with RepRap 3-D printers. Two different printer designs, a Cartesian [89] and a delta printer [90], were used to produce the parts out of 1.75 mm polylactic acid (PLA) filament. The printer design employed is ultimately irrelevant as both produce shapes using exactly the same method and materials and are different only in the way the print head is moved. Both printers were equipped with hot ends having 0.5 mm nozzles and prints were sliced at a layer height of 0.25 mm and a print speed of 60 mm/s. Parts were printed in plates, that is all of the printed parts needed to assemble a syringe pump were printed in one printer cycle.

RepRap printers typically interface with a host program running on a computer but can also run independently, reading g-code stored on a memory card. The syringe pump was printed using a host computer running ReptierHost [91] another freely available, open-source software written specifically for RepRap and RepRap-like 3-D

printers³. For detailed instruction on the construction and operation of the printers see [89, 90].

4.4 Syringe Pump Control and Interface

The syringe pump is controlled by an open-source Python program developed here [87] running on a Raspberry Pi, which is an ARM based computer running GNU/Linux [29, 92]. The Raspberry Pi is an inexpensive, credit card-sized computer having integrated networking, sound, video, USB host and most importantly, exposed and readily accessible I/O lines. The wiring diagram for the syringe pump controller (Figure 4.2) utilizes a single Pololu A4988 stepper controller, which controls the stepper motor that drives the syringe pump. The Raspberry Pi is installed with the standard Raspbian operating system [92]. A custom web server is run, which serves a web page via either wired network or wirelessly via a wireless USB adapter attached to the Raspberry Pi's USB port. Any computer on the network can then control the pump through this web page (Figure 4.3).⁴

4.5 Calibration and Performance Assessment Methodology

The pump is calibrated by setting it up with an initial calibration value set to 1 mL/mm. A small arbitrary volume appropriate for the size of syringe used is pushed twice from the syringe and the actual value of the second push is measured. This is done to partially account for drops staying on the end of the syringe. This number is divided by the amount the syringe was told to push out, the resulting number goes into the calibration window. The sequence is repeated three times to ensure correct calibration.

³Since the publication of this paper, the author of Repetier Host made the program closed source, but there are many open source options available. One of those options is Franklin, which is described in Chapter 2.

⁴Figure 4.3 contains a screenshot of the interface in a web browser. Window decorations of the Iceweasel web browser and Gnome desktop environment are used under the Fair Use exception of copyright law. Fair Use is applicable, because the illustration is similar to a quote, both Iceweasel and Gnome are freely available to everyone, the screenshot is an insignificant part of the programs and reading this dissertation is not a substitute for using those programs.

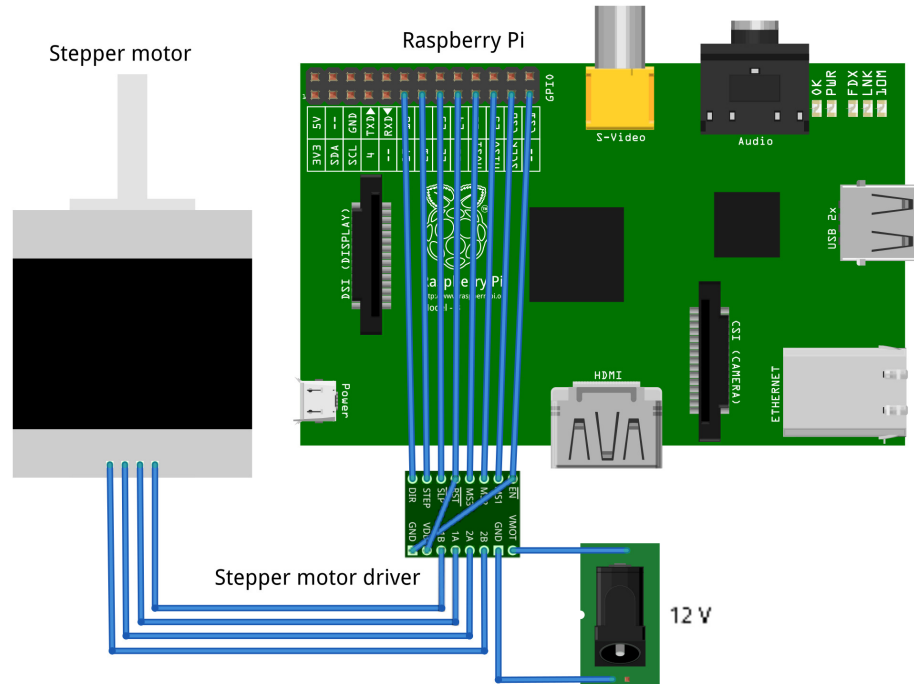


Figure 4.2: Electronic schematic of Open-Source Syringe Pump.

The force produced by the lead-screw actuated design was measured by placing the assembled syringe pump with a steel rod in place of the syringe in a frame along with a 30 kg-capacity scale. The pump was oriented such that the motor end sat upon the scale and the steel plunger faced upward, pressing against a fixed platform. The pump motor was advanced until it stalled or a component failed and the maximum force produced was read off the scale display.

The pump's maximum delivery rate is a function of the speed at which the motor stalls. Stall speed was determined by increasing pulse rate to the motor until it stalled and then decreasing to the point where it ran again, establishing the maximum speed and therefore maximum delivery rate.

Precision was tested by repeated delivery of a preset volume (fixed by setting the total number of motor steps) of distilled water onto a Mettler AE100 scale having a readability of 0.1 mg. The relative humidity within the weighing chamber was maintained in a saturated state by placing containers of distilled water in it, permitting it to equilibrate and then ensuring that it was kept well sealed for the duration of the assessment. Performance of both the NEMA11 and NEMA17 pumps was assessed at different microstepping settings.

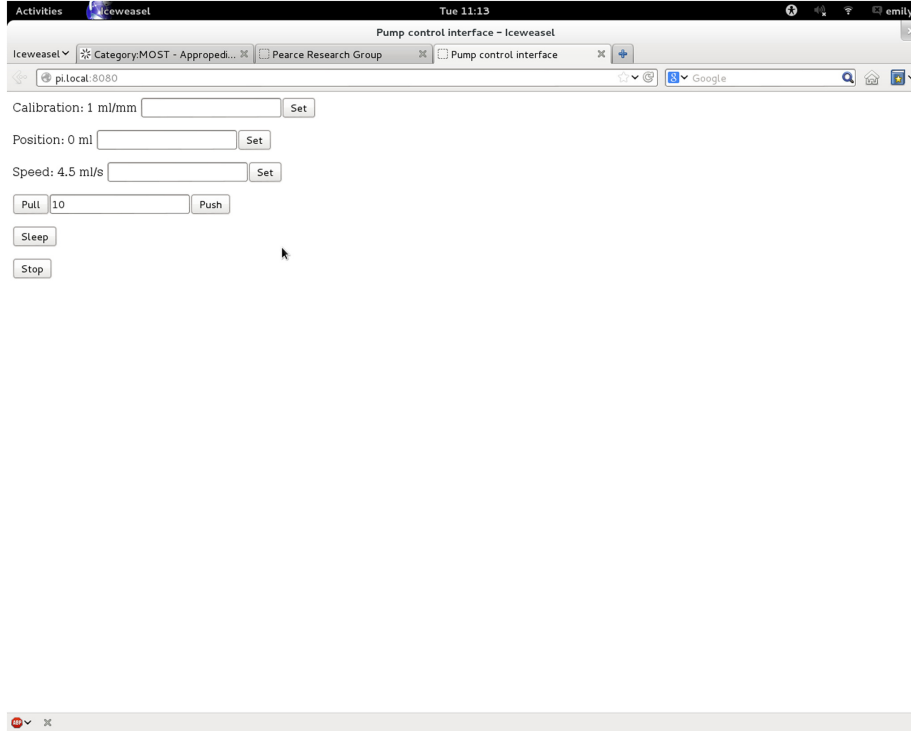


Figure 4.3: Screenshot of Syringe Pump Web Interface.

4.6 Results

Three different pumps were assembled, all of which are relatively easy to construct from the parts shown in exploded view in Figure 4.4. Assembled pumps are shown in Figures 4.5, 4.6 and 4.7 for the Nema 11, Nema 17 and dual Nema 17 pumps, respectively. The dual version consists of two identically sized pumps connected in parallel to the motor controller (Figure 4.8)⁵. The pumps are driven synchronously at the same rate. The controller has the capacity to drive more than one pump simultaneously if required. If one of the connections from the connector to the pump shown in Figure 4.8 is reversed then the two pumps will go in opposite directions.

The force developed by the OS syringe pump depends on the motor used. When pushing on an immovable object, the NEMA17 version produced 200 N, and the NEMA11 version produced 93 N. Force did not appear to be affected by the microstepping rate, probably due to the nature of lead-screw actuation that is inefficient at translating force into rotation, particularly with the thread pitch and profile used in this design.

⁵Both serial and parallel connections are possible. The figure shows the motors connected in series.

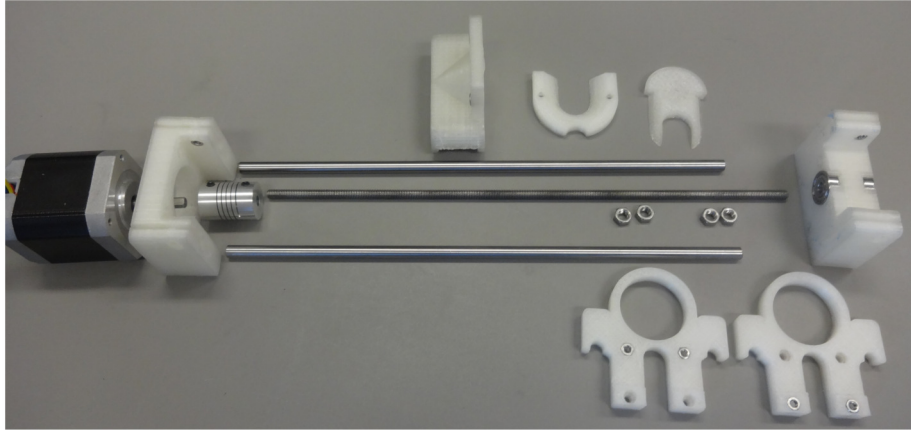


Figure 4.4: Exploded view of Open-Source Syringe Pump.

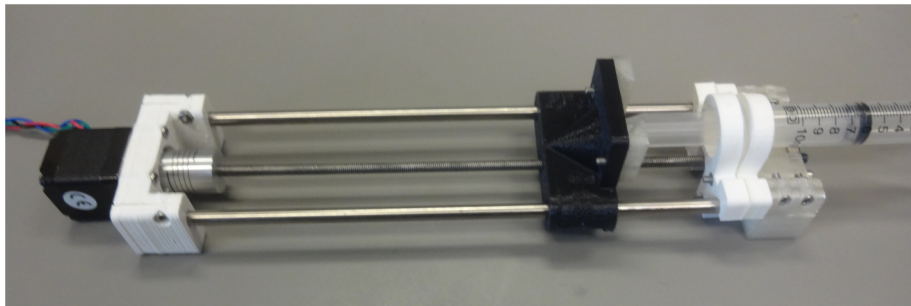


Figure 4.5: Digital Photograph of Open-Source Syringe Pump version Nema 11.

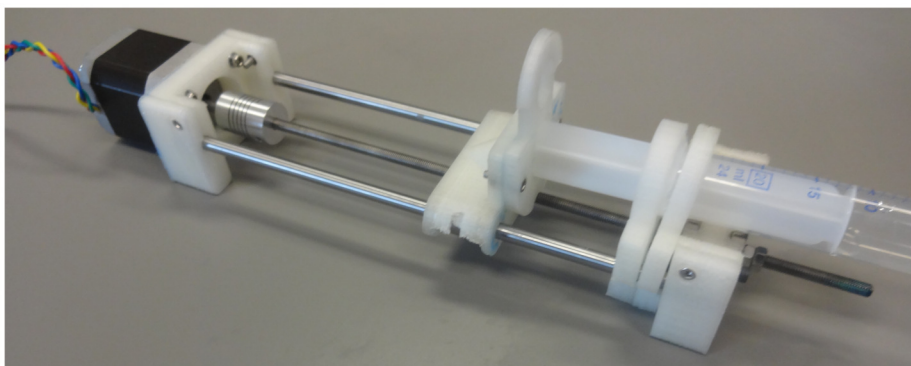


Figure 4.6: Digital Photograph of Open-Source Syringe Pump version Nema 17.

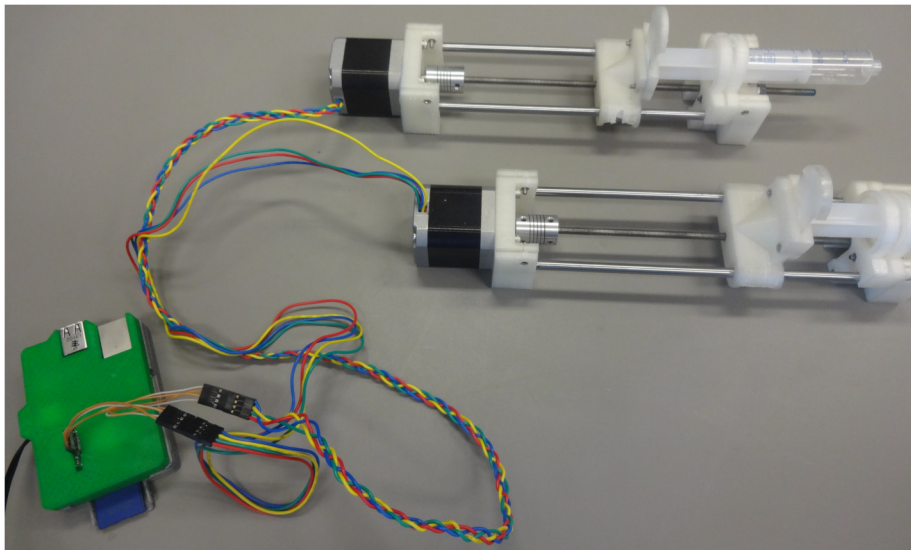


Figure 4.7: Digital Photograph of Open-Source Syringe Pump version of the Dual Nema 17 Pump.

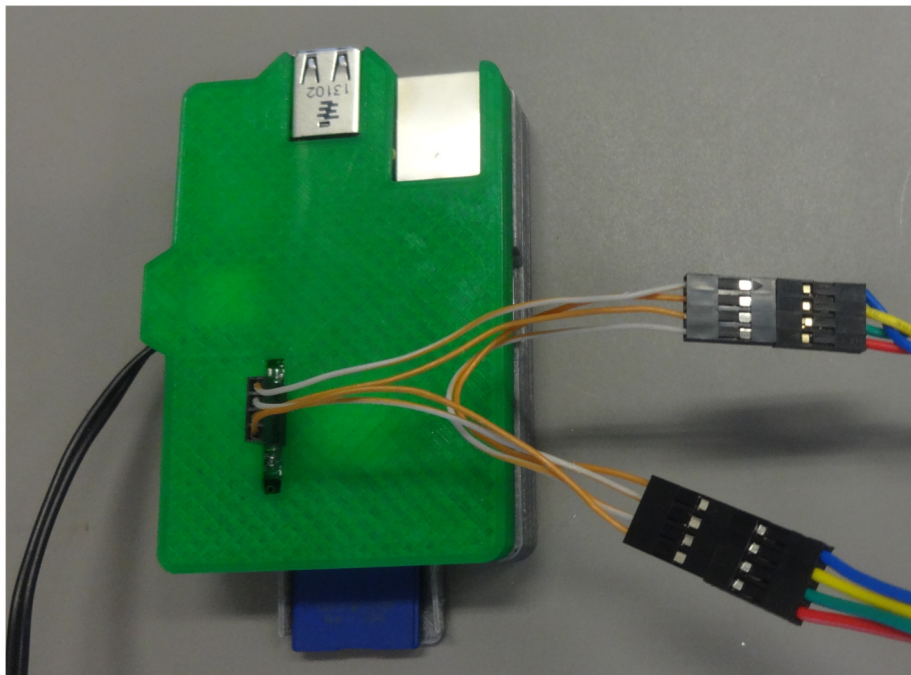


Figure 4.8: Digital Photograph of the Dual Pump connection.

Table 4.2

Coefficient of variation as a function of microstepping for NEMA 11 and NEMA 17 open-source syringe pumps.

	Microstepping		
	1	4	16
NEMA11	1.17%	0.56%	1.55%
NEMA17	3.66%	2.13%	2.26%

The force developed was sufficient to cause damage to some of the printed parts, none of which resulted in impairment and new parts could quickly be printed and replaced.

When calibrated with a 25 mL syringe, the NEMA17 version yielded a maximum delivery rate of 2.1 mL/s and the NEMA11 version yielded 1.4 mL/s when calibrated with 10 mL syringe. Accuracy was +/-1% for the NEMA11 and +/-5% for the NEMA17 measured in 1 mL increments. Precision was found to be relatively insensitive to microstepping for both the NEMA11 and NEMA17 (Table 4.2). The coefficient of variation when delivering approximately 1 mL of distilled water was about 3% or less regardless of microstepping and it is very likely that precision is actually better than reported as the measurement method was limited to the volume of a single drop (e.g. ≈ 20 micro Liters). It is unlikely that microstepping need be employed as the 200 step/revolution motors coupled with a properly sized syringe should provide virtually any resolution demanded.

It is clear that using open-source methods reduced the cost of the pumps considerably from commercial pumps as summarized in Table 4.3. The single syringe pumps have a part cost under \$100 using hardware from online retailers. This includes the Raspberry Pi controller that permits control of the syringe pump from virtually every web enabled device available. Commercial syringe pumps can cost anywhere from \$260 to over \$5000 as seen in Table 4.3.

Overall, using completely open source methods, this pump is economical, user friendly, and accurate. Even considering the approximate \$500 price of the RepRap 3-D printer, the value of this approach to design and manufacturing far exceeds that of commercial units, particularly for resource starved laboratories.

Table 4.3

Specifications for the open-source syringe pump are shown compared to commercial pumps.

Name	Price (US\$)	Speed (mL/s)	Size (mL)	Force (N)	Acc. (%)	Repr. (%)
MOST OS Syringe Pump NEMA11	90	1.4	10	93	1	0.6
MOST OS Syringe Pump NEMA17	97	2.1	25	200	5	2.1
MOST OS Dual Syringe Pump NEMA17	154	2.1	25	200	5	2.1
NE-300 “Just Infusion” syringe pump	260	0.417	60			
B.Braun/McGaw BD 360 Syringe Pump	435	0.1			3	
GenieTouch Syringe Pump	675	3.68		169.13		
NE-4000 Programmable Dual Syringe Pump	928	2.0	60	444.82	1	
Med Associates PHM-111EC	1343	0.119				
Fusion Touch 400 Syringe Pump	1350	0.167	10	222.41	0.35	0.05
Fisher Scientific Single Syringe Pump	1509	0.144	60			
Sono Tek Syringe Pump	1800	0.5	60			
Cole Parmer Dual Syringe Infusion Pump	2606	2.45	140		0.5	0.2
Cole Parmer Continuous Flow Syringe Pump	3947	1.17	60	177.93	1	0.1
Gilson 402 Syringe Pump	5000-5500	2	25		1.8	0.8

4.7 Discussion

As has been demonstrated previously, the open-source ecosystem lends itself well to research endeavors, especially with regards to maximizing the value of a research dollar [10, 59, 93, 94]. This is particularly true when designs for desired components, or even designs for similar components to those desired, are made freely available for use and customization [10, 59, 65]. Armed with open-source 3-D printers and hardware, freely available and open-source software and designs, researchers can design and manufacture bespoke apparatus at a small fraction of the price of commercial offerings. The ability to alter and tune designs to produce apparatus that better align with research goals eliminates “making do” with what is available commercially. By way of example, this paper presents an elegantly simple design for a syringe pump that performs admirably and should serve as a good foundation for derivation of better and more useful apparatus for specific research goals.

The simplicity of the design coupled with ready access to its source makes it very easy to customize and construct; even first year students with limited exposure to such activity are able to assemble a complete, working system. The cost of the entire system, including the controller and web-based control interface, is on the order of 5% or less than one would expect to pay for a commercial syringe pump having similar features and performance. The platform is not limited to just use as a syringe pump; it is a relatively high precision linear actuator that can easily be modified for use for positioning, i.e. for stages for microscopy. Similarly it could be used as a head for 3-D printing with viscous media. 3-D printing and liquid handling with a syringe pump could be combined as has been done recently by Kitson et al., to produce user-friendly reactionware for chemical synthesis and purification [63]. Using open-source RepRap 3-D printers and the open-source syringe pump developed here chemists not only have complete control over every aspect of hardware, but can also set up the experiments for a fraction of the cost of commercially available tools.

Incremental improvement of designs in the open-source ecosystem tends to occur organically [59]. It is therefore reasonable to expect that as the population of the interested audience grows, the rate of innovation increases, perhaps at a much greater rate than could be expected in commercial R&D centers. This incremental approach to development not only takes place at a rapid pace, it spreads the cost of development over the entire user/developer community with the currency being predominantly the time spent by the individual developers. Since freely available open-source designs can be made available to the entire globe, even small time investments in development can have significant impact. This is especially important given that the tools developed can be considered appropriate technology and are of particular interest to poorly

funded laboratories such as those in undeveloped and developing economies [63]. Development of open-source designs can, from that perspective, be considered a form of philanthropy, although the developer also benefits by the product of his work and the improvements made to it by others [59].

The design presented here is deliberately simple; it is intended to demonstrate the utility and efficiency of the open-source method of development and provide one starting point for derivation of improved designs. There is (by design) ample opportunity for improvement and future work. A syringe pump can be used for a variety of applications requiring carefully controlled dosing of reagents, pharmaceuticals, delivery of viscous 3-D printer media, etc. All of these applications have specific requirements that this endlessly customizable design can be tailored to meet. For instance, microstepping may not be required, making a less expensive motor controller suitable and driving the cost of the syringe pump even lower.

4.8 Conclusions

An open-source and freely available design for a simple to build and customize syringe pump has been provided and working pumps have been constructed and evaluated. The design performs well as compared to much costlier commercial models while permitting virtually endless customization and so should suit the needs of a given research activity requiring a syringe pump. Only readily available, open-source hardware and software were used for the design and manufacture of the pumps, further validating application of open-source methodologies for development of research-ready laboratory equipment.

4.9 Acknowledgments

The authors would like to acknowledge helpful discussions with P. Fraley.

Chapter 5

Free and Open Source Automated 3-D Microscope

Abstract¹

Open-source technology not only has facilitated the expansion of the greater research community, but by lowering costs it has encouraged innovation and customizable design. The field of automated microscopy has continued to be a challenge in accessibility due the expense and inflexible, non-interchangeable stages. This paper presents a low-cost, open source microscope 3-D stage. A RepRap 3-D printer was converted to an optical microscope equipped with a customized, 3-D printed holder for a USB microscope. Precision measurements were determined to have an average error of 10 μm at the maximum speed and 27 μm at the minimum recorded speed. Accuracy tests yielded an error of 0.15%. The machine is a true 3-D stage and thus able to operate with USB microscopes or conventional desktop microscopes. It is larger than all commercial alternatives, and is thus capable of high depth images over unprecedented areas and complex geometries. The repeatability is below 2-D microscope stages, but testing shows that it is adequate for the majority of scientific applications. The open source microscope stage costs less than 3% to 9% of the closest proprietary commercial stages. This extreme affordability vastly improves accessibility for 3-D microscopy throughout the world.

¹The material contained in this chapter has been accepted for publication in the Journal of Microscopy.

5.1 Introduction

The free and open source software (FOSS) community has demonstrated that by facilitating participation in technical projects with little to no startup costs, meaningful contributions from the community can be made [95, 96, 97]. FOSS is also referred to as libre software to emphasize development founded on freedom as opposed to price. This scaled collaboration results in superior design with lower associated cost due to the continuous improvement in software code, thereby making it more robust and innovative [95, 98]. FOSS has shown to be more reliable and relevant to users as they are co-developers [99]. For instance, 97% of the world's supercomputers operate on FOSS GNU/Linux [100]. The success of open source software has provided an alternative to expensive and proprietary systems by allowing for reduced research and development costs [96] as well as more flexible design [101]. Open source development outside of purely software has established a particularly vibrant 3-D printing community around the self-replicating rapid prototyper (RepRap) 3-D printer family [2, 3, 4, 93, 102]. FOSS and open hardware design can be combined with RepRap 3-D printing for distributed digital fabrication of low-cost scientific equipment [10, 103] including: colorimeters [66], nephelometers [36], turbidimeters [104], liquid autosamplers [105], microfluid handlers [106], biotechnological and chemical labware [107, 108, 109], mass spectroscopy equipment [110], automated sensing arrays [111], phasor measurement units [112], optics and optical system components [65], DNA nanotechnology lab tools [113] and compatible components for medical apparatuses [114]. The open sharing of digital design has reduced capital cost to an unprecedented 90-99% decrease from the cost of conventional equipment [59, 115]. The lateral scaling of shared design has created substantial value [116] resulting in hundreds and even thousands of percent return on investment for science funders [117]. This open source development methodology holds some promise for improving accessibility of automated microscopy.

The field of automated microscopy faces a number of challenges in becoming more accessible to those without extensive training or funding. Automated microscope stages are typically costly and limited in use due to the lack of adaptability. However, the quality of digital sensors has significantly improved, combined with a decrease in cost, which make the option of using an open source approach technically possible. Furthermore, the associated software, traditionally a proprietary component of automated microscopy, has begun to transition into open source programs such as ImageJ and iMSRC. Not only has automated microscopy aided in improving the reproducibility of results in the greater research community, but it has allowed more individuals to learn proper operation and perform more complex analysis techniques.

Although advances have been made in creating inexpensive microscopes [118, 119],

established automated microscope models remain essential for scientific research such as the detection of tumor cells in bone marrow [120], pollen analysis [121], and three-dimensional examination of cellular structures and macromolecules [122]. The cost of sophisticated instrumentation is in the thousands or tens of thousands of dollars, creating a significant barrier to joining the scientific community in those fields. Decreasing that cost not only enables scientists with limited funding and resources to perform higher quality research, but it allows everyone in the scientific community to allocate funding to other needs [115]. Open source space has provided a new platform of innovation for automated microscopy [119]. By taking advantage of existing tools such as Raspberry Pi, Arduino controllers, 3-D printed parts and beam structure, advancements can be made toward creating a generic system of ultimately customizable automated microscopes [119]. In addition, the RepRap 3-D printer has been shown to be a practical low-cost scientific 3-D stage [123].

This study looks to build on the ongoing developments in automated microscopy in the open source community. A RepRap 3-D printer was converted to a 3-D microscope, and as this study is released as free and open source, all interfaces are readily available and can be easily integrated into new or existing experimental setups. The 3-D microscope is validated using experiments to compare its accuracy and precision to commercial alternatives. In addition, the system is demonstrated for photo stitching and focus stacking. Finally, the open source 3-D microscope is compared to proprietary commercial tools and the results are presented and discussed.

5.2 Background

Automated microscope stages allow for examination of multiple specimens by facilitating stage movement in the x, y and z directions. Stages are typically for either upright or inverted microscopes, however as they are not adaptable for both and are frequently designed for specific microscope models, they have limited practical use. Stage sizes range from 135 mm x 85 mm to 275 mm x 175 mm, and both repeatability and reproducibility are less than one micron. Stages of these parameters, however, can cost the user anywhere from US\$4,900 [124] to US\$16,000 [125]. This cost does not include additional costs associated with the software package and hardware required to mount the stage on the microscope.

The field of automated microscopy has seen a number of recent developments, in part due to the advancement of digital sensors. The increase in quality coupled with the decrease in cost has played a significant role. Automated microscopy contributes in

a meaningful way to the greater research community by making results more reproducible [126], enabling less-trained operators to obtain quality measurements [127], and facilitating more complex analytical methods such as neural-based pattern detection [128]. While the software used to control automated microscopes and analyze measurements has traditionally been proprietary, free and open source options have since matured. iMSRCS is a software developed in recent years that combines flexibility with ease of use in order to enable various applications [129]. ImageJ is another software tool that allows for automation of experiments with the help of scripts and plugins [130]. It can also be combined with MicroManager to control an automated microscope [131].

5.3 Experimental

A delta RepRap derived from the MOST delta, called the Athena, is robot with exchangeable tools that can be used as a 3-D printer [22, 23]. Smaller tools can be mounted on the end effector, enabling movement over a static object. The platform can also be mounted to the end effector, allowing for bigger tools such as a metal welder to be mounted in a stationary position on top of the machine [23]. For use in microscopy, a small USB microscope can be mounted as a mobile tool or a heavy microscope can be mounted as a stationary tool.

The RepRap was controlled using Franklin [18] which allows external scripts to control the machine, thus allowing integration of the microscope in a larger experimental setup [1]. For example, the microscope can be moved vertically to keep a moving object in focus, or horizontally to keep it in the view. A script that is available at [18] enables a game controller to move the tool. This allows the user to control the microscope without looking at the keyboard or screen, instead providing a full screen view of the microscope output. Four positions can be stored and linked to buttons on the game controller for ease of revisiting.

Several experiments were performed in order to demonstrate the capabilities of the device. To test the precision of the positioning for long moves, a calibration slide with a dot of 70 μm diameter was positioned at two positions on the platform; one in the center at (0,0) and one at (0,100). The microscope was then moved to a different location at the edge of the range before being repositioned to the calibration dot. The position of the dot in the microscope image was recorded each time. The measurement was performed using 36 points on the edge of the range, which were equally spaced with a spacing of 10 degrees. Every position was measured at three speeds: 2, 20 and 60 mm/s.

To test the precision of the positioning for short moves, the microscope was focused on the dot and moved over a circle with a radius of $50\ \mu\text{m}$ in steps of 10 degrees. These steps were small enough to ensure that the dot remained in view of the microscope. The dot position was recorded after every move, and compared to the target values. A statistical analysis was then performed on the data. For each recorded image, the position of the dot was determined. For each dot, the distance and direction of that dot compared to the average of all dots was determined.

To test the accuracy of the device, a Vernier caliper with a precision of 0.005 mm was used to measure the length between the corners of a piece of copper. The microscope was then moved to position each of the two corners at the same position as noted in the recorded image. The associated movement required was measured and compared to the actual length of the edge.

The setup of the experimental procedures was flexible due to the ability to script Franklin. To demonstrate this, two experiments were automated. First, an object with significant height was placed on the platform and images were recorded at several heights. Those images were combined into a single image with photo stacking software [132]. Second, a large flat object was placed on the platform and images were recorded at different horizontal positions with some overlap. Those images were combined into a single image with photo stitching software Fiji, which is imageJ with many plugins pre-configured [133].

The full bill of materials and SCAD files for the 3-D printed parts are released under and open source license here [134].

5.4 Results and Discussion

5.4.1 Results

The fully assembled instrument is shown in Figure 5.1, with the lead screws, anti-backlash springs, endstops and USB microscope labeled. A holder was designed in OpenSCAD [86] and 3-D printed as shown in Figure 5.2. This holder was then supplemented with ball bearings in order to accommodate a USB microscope. This capability highlights the ability to three-dimensionally print custom parts on the same machine adapted for optical microscopy.

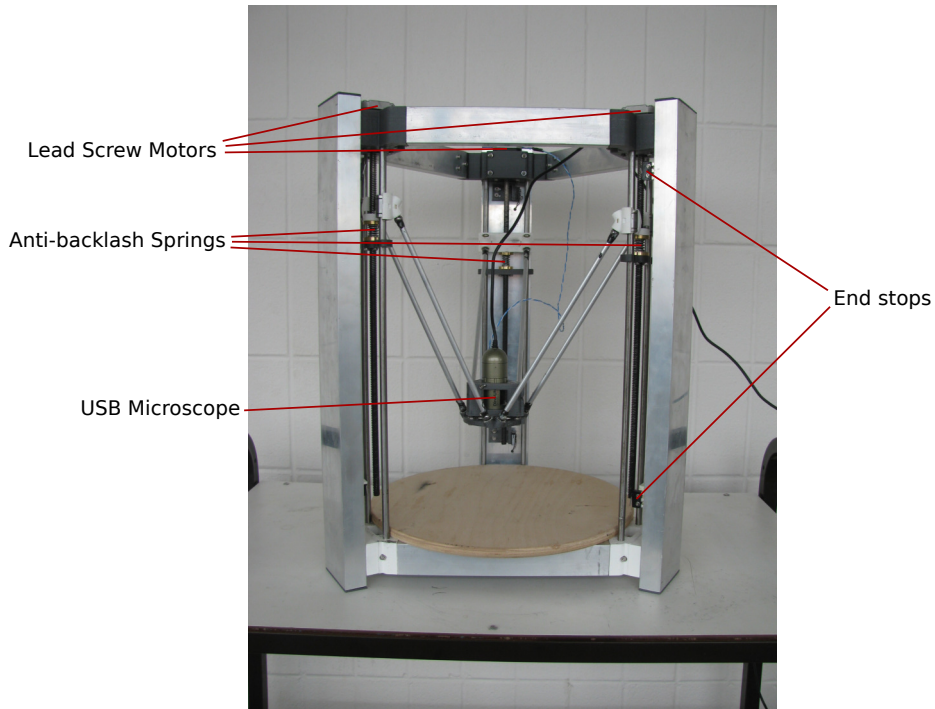


Figure 5.1: The open-source 3-D microscope stage with USB microscope.

The microscope can also work in mobile sample mode (or stage mode) by removing the tie rods from the down position and placing them in the up position on the carriages. Stage mode moves the sample underneath a conventional microscope as shown in Figure 5.3.

The 3-D microscope stage electronics is comprised of a Beaglebone Black and Melzi (Arduino compatible) board. The electronic schematic for the system is shown in Figure 5.4. The three stepper motors that create the movement of the 3-D microscope are controlled by the Melzi, which is itself run by the Beaglebone Black. A computer is used to directly control the USB camera. This computer is also connected via the network to the Beaglebone Black. The machine is similar to the open-source multi-material additive and subtractive MOST delta RepRap [22] and the more advanced open source derivative called the Athena [135]. The main differences between this machine and those are that wood support is replaced by aluminum tubing and that the carriages are not moved by belts, but by lead screws with anti-backlash nuts. Detailed build instructions for the Athena are available on-line [135] and can be used as a guideline for building this machine.

The 3-D microscope movement is controlled by Franklin, the interface of which is

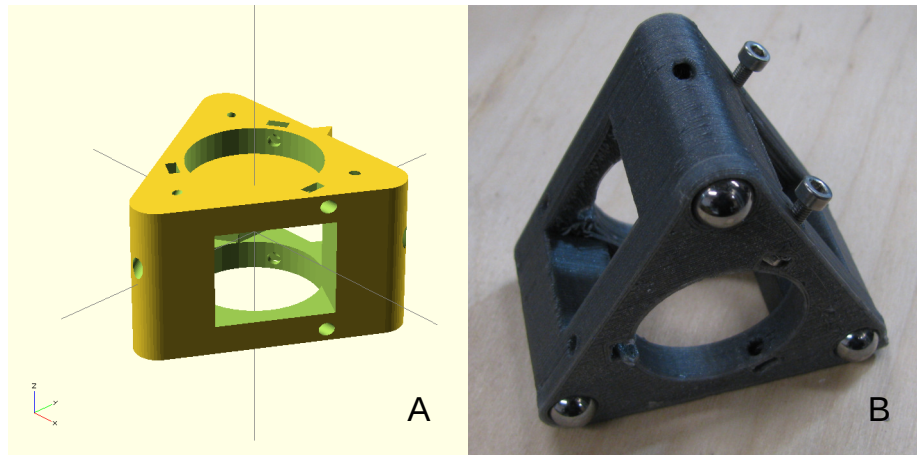


Figure 5.2: A) OpenSCAD STL rendering of USB microscope holder, B) Assembled USB microscope holder with ball bearings and screws.

shown in Figure 5.5.² It runs on the Melzi and the Beaglebone, and is accessed through a web browser. In the top left is a list of available G-Code to execute. Below that are the controls for moving the microscope. In the top right are control buttons and below that is a graph for all the temperature sensors; that is used when Franklin is controlling a 3-D printer. For the microscope, this graph is empty. All setup, configuration and calibration is also performed from this interface.

Photostacking while moving the microscope in the z plane is demonstrated in Figure 5.6, where Figure 5.6a shows a flower with the bottom in focus and 5.6b with the top in focus. These were part of a stack of 10 images. Figure 5.6c shows the final image of the stacking technique results in an image of greater depth than any of the original images. The result is thereby of greater value to the researcher.

Figure 5.7 demonstrates photostitching 18 individual images while moving the microscope in the x-y plane on a composite paint. The techniques demonstrated in Figure 5.6 and 5.7 can be combined to image large 3-D objects with significant depth.

The precision of the instrument was found to be dependent on the speed of movement. At the maximum speed of 60 mm/s, an average error of 10 μm was found. At the lowest measured speed of 2 mm/s, this average error was determined to be 27 μm .

²Figure 5.5 contains a screenshot of the Franklin interface in a web browser. Window decorations of the Iceweasel web browser are used under the Fair Use exception of copyright law. Fair Use is applicable, because the illustration is similar to a quote, it is freely available to everyone, the screenshot is an insignificant part of the browser program and reading this dissertation is not a substitute for using Iceweasel.

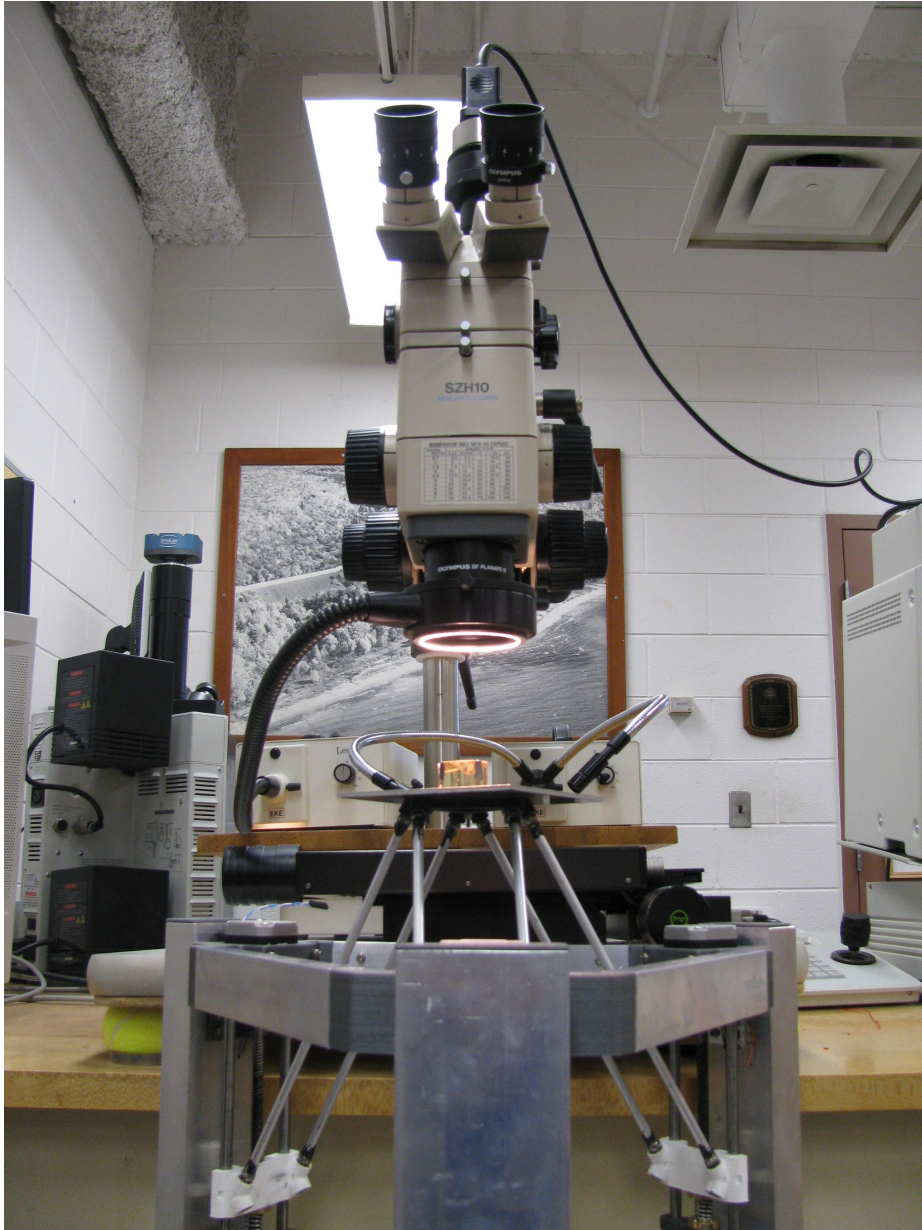


Figure 5.3: The 3-D microscope stage shown adapted to a stage. The stage is viewed from above by a conventional microscope.

This error is likely due to the lagging of components. At higher speeds, a sudden stop in movement presses the parts into their final position, resulting in a smaller positioning error. This conclusion was confirmed by the results of the small circles experiment that demonstrated that the actual position was slightly behind that of the target, resulting in a circle of smaller radius than instructed.

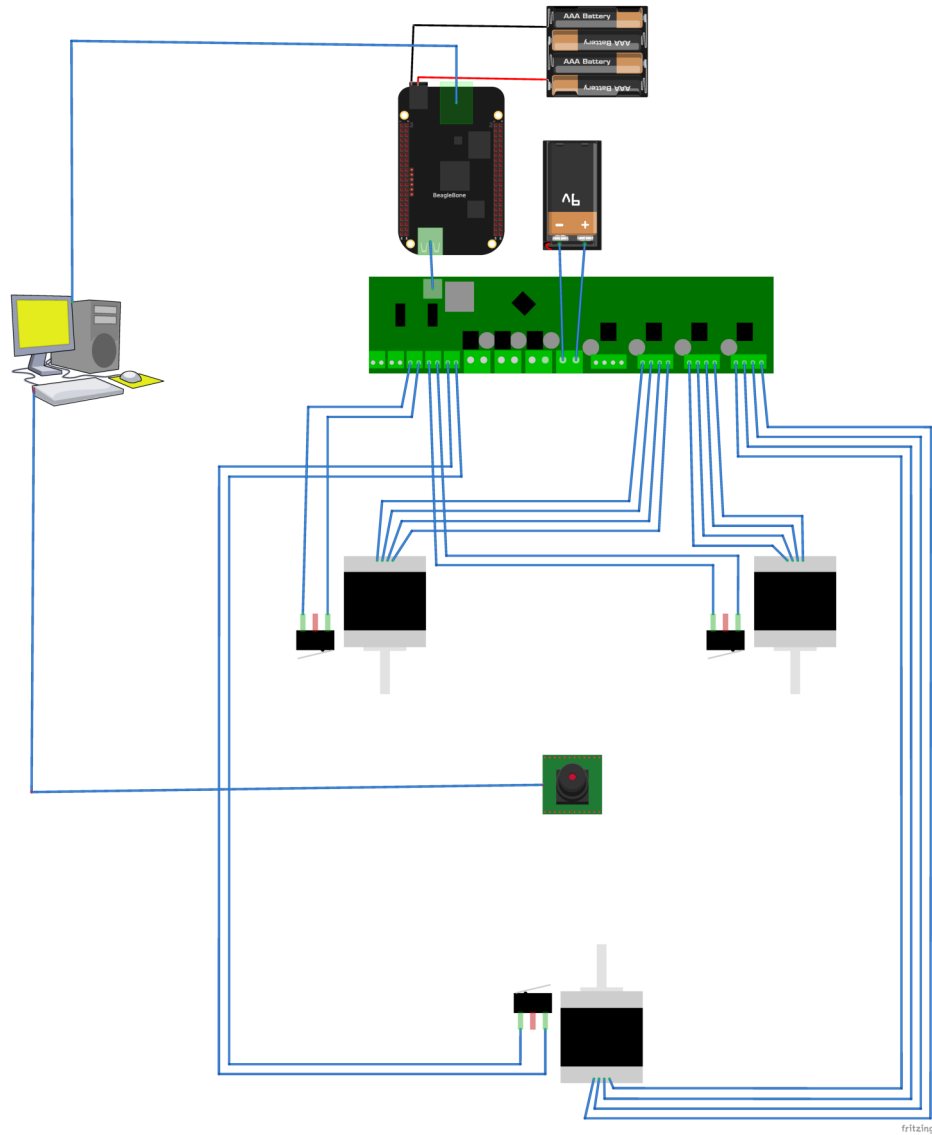


Figure 5.4: The electronic schematic of the 3-D microscope.

In measuring the accuracy of the device, a piece of copper was initially measured to be 101.22 mm long. The microscope measured a length of 101.64 mm, yielding an error of 0.15%. It should be noted, however, that no attempt was made to calibrate the machine for scale. The absolute error at long distances can be made just as small as the error associated with precision if a calibration is performed.

Most commercially available microscope stages for scientific applications are two-dimensional. They cost around \$10,000, have a repeatability between 0.2 and 2 μm , and they have a precision between 5 and 500 nm. The device presented here is

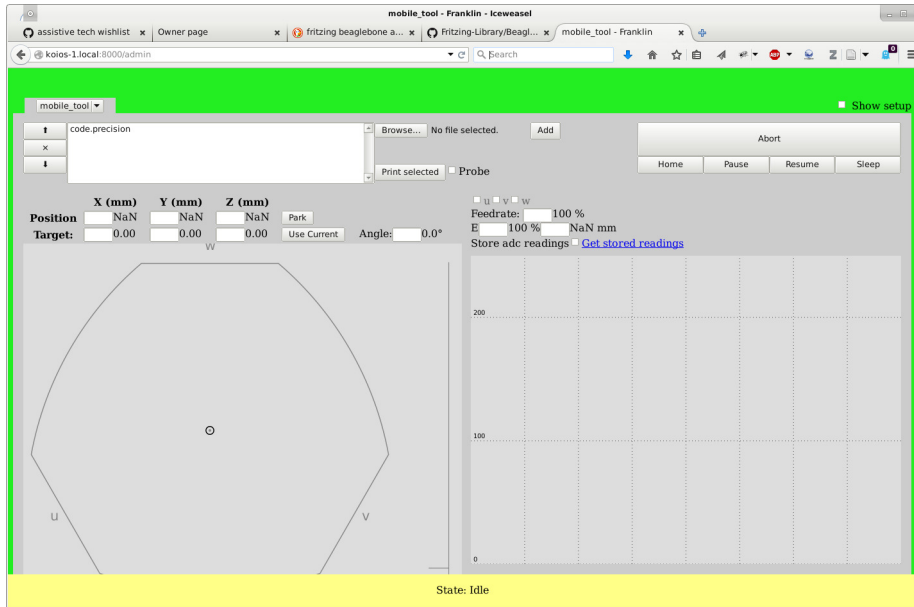


Figure 5.5: The Franklin interface allows users to control the movement of the USB microscope in x, y and z directions. Furthermore, G-Code files can be uploaded and executed.

three-dimensional, costs less than \$1,000, and it has a repeatability of $156 \mu\text{m}$ and a precision of less than $30 \mu\text{m}$. While commercial alternatives demonstrate superior operative standards, the experimental operations outlined here prove this new system to be sufficient for many scientific applications. The economic analysis of a proprietary machine compared to the OS variant described is summarized in Table 5.1.

5.4.2 Discussion

It should be pointed out here that the open microscope stage is a true 3-D stage. This is extremely rare for commercial stages as manufacturers expect the microscope to move in the Z direction, although this is not always an option. Because of this, there were no true commercially comparable equivalent products found on the market. Despite this limitation for comparison, there are several points that are clear from Table 5.1. First, the open-source microscope stage is larger than all commercial tools in the survey of proprietary equipment. The size of the open source stage was largely dictated by the size of the MOST delta from which it was derived. This obviously has clear advantages for imaging large objects with complex and large variations in geometries. These types of objects are simply not generally available for microscopic

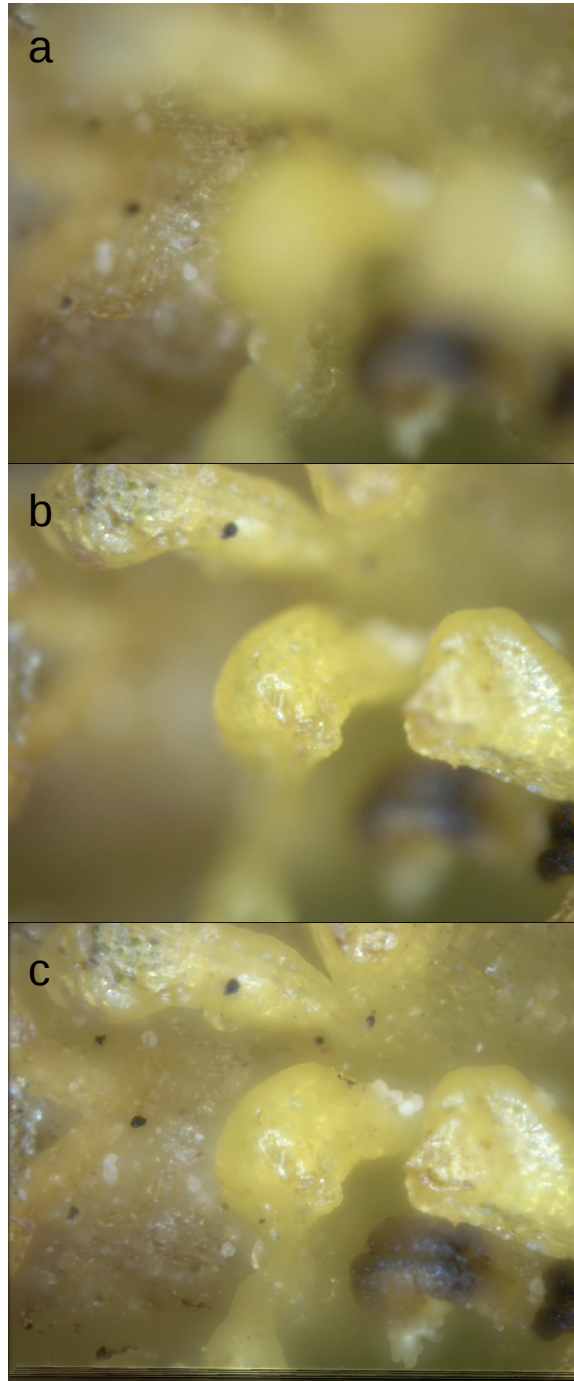


Figure 5.6: a) An image of a flower where the bottom is in focus. b) An image of a flower where the top is in focus. c) The final image of the stacking technique results in an image of greater depth than any of the original images.

Table 5.1
Comparison of open source microscope to proprietary microscope specifications.

Product Name	Price (US\$)	Travel Range (mm)	Max v (mm/s)	Rep. (μm)	Res. (nm)	Source
MOST Open Source Stage	425	$\varnothing 250 \times 200$		156	30,000	
MS-9400 XY Automated Stage	9,250 +5,220	225×100	7	0.7	100	[136]
MLS203-1 XY Stage	6,799 +2,959	100×75	250	0.25		[137]
Motorized XY Microscope Stage	4,900	100×120	85	2		[124]
Leica LMT260 XY Scanning Stage	16,000	120×80	500	0.25	5	[125]

imaging without disassembly or destruction (e.g. via mechanical slicing). Combining the demonstrated functionalities of the stage for both focus stacking (Figure 5.6) and image stitching (Figure 5.7), this device makes it possible for the first time to develop large composite images of the exterior of objects with enormous depth over a volume of cylinder measuring 250 mm in diameter by 200 mm in height. Future work could improve the usability of this tool by automating the focus for such stack-stitched images.

The costs of the open source stage are also at least an order of magnitude less than the proprietary commercial tools with even remotely similar functionality. The maximum speed of the open source stage is slower than the majority of tools, but is not the lowest as seen in Table 5.1. Speed may become important for large dynamic image capture. Finally, the repeatability, which was measured in this study to be over 150 microns was significantly higher than all the commercial devices. For the vast majority of microscopy applications this is not a problem as the user has visual feedback (e.g. the feature of interest is in the field of view). The one application, where a small error in repeatability is necessary is when doing photostitching with no overlap after performing a raster over a large area. As this study has demonstrated in Figure 5.7, use of open source photostitching software is an adequate solution as long as a small overlap is tolerated. The slightly slower resultant image taking time would only be a problem in experiments where the sample exhibits significant changes during the measurement, but such experiments are unsuitable for using photostitching without overlap as well.

For those needing improved repeatability the precision of the open source stage can

be improved using several approaches including decreasing the volume and using larger linear bearings to decrease unwanted movement in the carriages. In addition, higher pitch lead screws would lead to smaller steps which can be useful for high magnification work.

The work presented in this paper adds to the growing interest in open source tools for microscopy [65, 119, 138, 139, 140] demonstrates a variety of similarities to the open source microscope found on OpenLab Tools [119]. However, where that team aims to build a new machine entirely, this work targets currently existing and operational machines [124]. The benefit to this approach is that it negates the necessity of a new machine, and it expands the current applications of existent lab facilities. This limits the options for the optics that can be used in mobile tool mode compared to other microscopes, including the OpenLabTools open source microscope. This limit can be avoided when using the machine in stage mode, but this limits the sample size.

This work opens the door to a multitude of potential future research investigations. The machine presented in this paper currently does not support Micro-Manager, a popular program for controlling microscopes in scientific experiments. Adding this functionality would further broaden the platform for open-source technology. Additional work could also be done in the area of calibration. As this system currently stands, calibration requires the user to understand the associated underlying processes. A separate calibration interface could be designed in order to automate and simplify calibration.

Finally, it should be pointed out that the extreme affordability of this automated 3-D microscope vastly improves accessibility. Few research groups, particularly in the developing world can afford more than \$10,000 for an automated microscope stage. This stage fits into the established paradigm of 3-D printing reducing research costs [17]. By enabling more scientists to have access to cutting edge tools such as this microscope, the number and diversity of the scientific community is expected to increase [115].

5.5 Conclusions

This paper presents a low-cost, open source microscope that maintains the use of the 3-D printer architecture on which the hardware is built. The machine is a true 3-D stage and thus able to operate with USB microscopes or conventional desktop microscopes. It is larger than all commercial alternatives, and is thus capable of high

depth images over unprecedented areas and complex geometries. The repeatability is below 2-D microscope stages, but testing shows that it is adequate for many scientific applications. The precision of the instrument was found to be dependent on the speed of movement (i.e. an average error of 10 μm , 60 mm/s and 27 μm at 2 mm/s) and accuracy tests yielded an error of 0.15%. The open source microscope stage costs less than 9% of the closest proprietary commercial stages. This extreme affordability vastly improves accessibility for 3-D microscopy throughout the world.

5.6 Acknowledgments

The authors would like to acknowledge technical support from G. C. Anzalone and 3D4Edu, LLC for equipment.

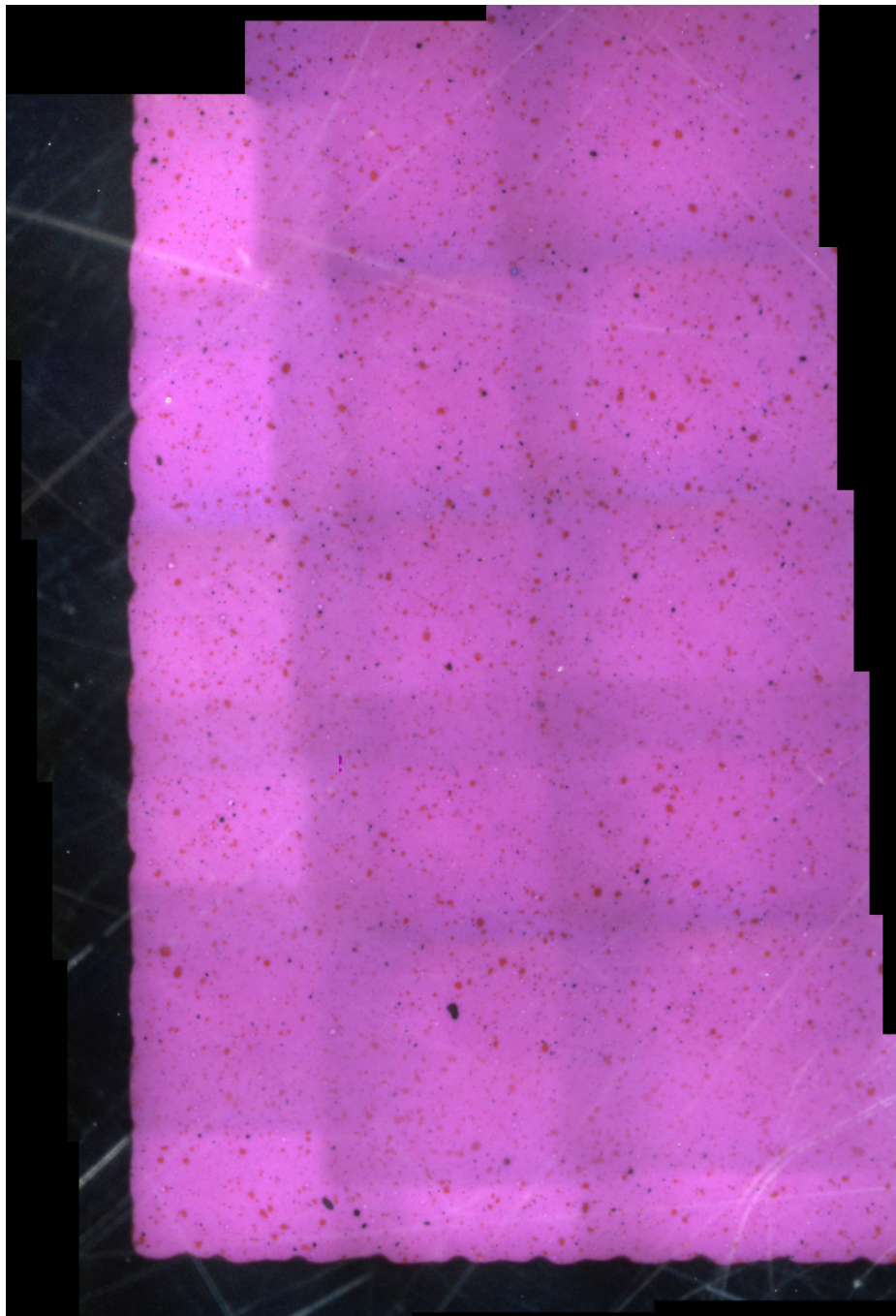


Figure 5.7: Final photostitched image of complex aggregate paint.

Chapter 6

Improved Model and Experimental Validation of Deformation in Fused Filament Fabrication of Poly Lactic Acid

Abstract¹

RepRaps (self Replicating Rapid prototypers), which 3-D print objects using fused filament fabrication (FFF), have evolved rapidly since their open source introduction. These 3-D printers have primarily been limited to desktop sizes of volumes of $\approx 8,000 \text{ cm}^3$, which has limited the attention of the scientific community to investigating deformation of common thermoplastics such as polylactic acid (PLA) used in FFF printing. The only existing physically relevant deformation model was expanded to use a physics-based temperature gradient instead of a step function. This was necessary to generalize the model to 3-D printing in a room temperature environment without a heated chamber. The thermal equation was calibrated using thermal measurements and validated by measuring curvature in printed objects. The results confirm that this is a valid model for predicting warpage of thin vertical walls of PLA. Additionally, the effect of annealing was examined. It was found that at a temperature of 50°C , no shrinkage or crystallization takes place, but at 90°C the PLA rapidly crystallizes to around 20% crystallinity. This indicates that heated bed temperatures should be maintained at 50°C or lower to avoid print failure (premature substrate

¹The material contained in this chapter is in preparation for submission to a journal.

release) with PLA. At 90°C, the annealing is accompanied by a 5% size decrease in both horizontal dimensions, but an 8% increase in the vertical dimension. Thus, the volume decreased by only 3%. This observation may lead to potential methods of improving slicing of printing large PLA objects with FFF.

Acknowledgments

The authors would like to thank Dr. Edward Laitila, Gerald Anzalone, Mark Klein for their help and discussion, and Thermoanalytics for the use of the IR camera.

6.1 Introduction

RepRaps (self Replicating Rapid prototypers) which 3-D print objects using fused filament fabrication (FFF) have evolved rapidly since their open source introduction [2, 3, 4]. There are now dozens of RepRap designs [141] and the majority of the prosumer 3-D printers on the market are derived from RepRaps [142] including Make magazine’s best 3-D printer of 2015 the open source Lulzbot Taz [143]. With this rapid technological evolution, FFF 3-D printing applications have also exploded and have opened many doors for applications that were previously only affordable for large companies such as: open source appropriate technology for sustainable development [16, 17, 144, 145], personal fabrication [146], small custom print shops [147], scientific equipment [10, 36, 59, 65, 103, 108, 123, 148], microfluidics [149], solar photovoltaic racking [150, 151], dentistry [152, 153], medicine [114, 154, 155, 156]; farming equipment [157], education [21, 158, 159, 160, 161, 162, 163], and museum replications [164].

So far, the new FFF 3-D printers have mostly been limited to desktop sized models, which primarily print in poly lactic acid (PLA) with a thermal coefficient of linear expansion α of $8.5 \cdot 10^{-6} \text{ K}^{-1}$ [165] because it demonstrates less warping during a print than other materials such as ABS (α up to $15 \cdot 10^{-6} \text{ K}^{-1}$ [165]) or HDPE (α up to $11 \cdot 10^{-6} \text{ K}^{-1}$ [165]) plastic and the emissions during printing are less dangerous [166]. Furthermore, PLA is made from corn-based resin, making it non-toxic, biodegradable, and able to be produced in environmentally friendly, renewable processes [167, 168]. The size of the printers (around $8,000 \text{ cm}^3$) have limited the size of the prints that can be made with them. There have, however, also been attempts to use larger machines as open source hardware 3-D printers such as the Gigabot ($212,400 \text{ cm}^3$) developed by re:3D [169].

Because of the limited adoption, not much attention has been given so far to the challenges that large scale FFF 3-D printing presents. The main challenge is to limit the deformation of the printed part during the printing process. To obtain deformation-free high-quality prototypes remains one of the most important challenges in the rapid prototyping field [170] as a whole and the additive manufacturing field in particular. Due to temperature gradients in the print, the edges tend to warp upwards as seen in Figure Figure 6.1. The effect scales with the size of the printed object, so with larger prints, it is a larger absolute deformation. If the effect is too large, the object will release from the substrate before it is completed, and the print will have failed. Residual stress [171] and crystallinity [172] are important parameters for the properties of printed objects and the determination of deformation, however the deformation process of FFF is not well understood.

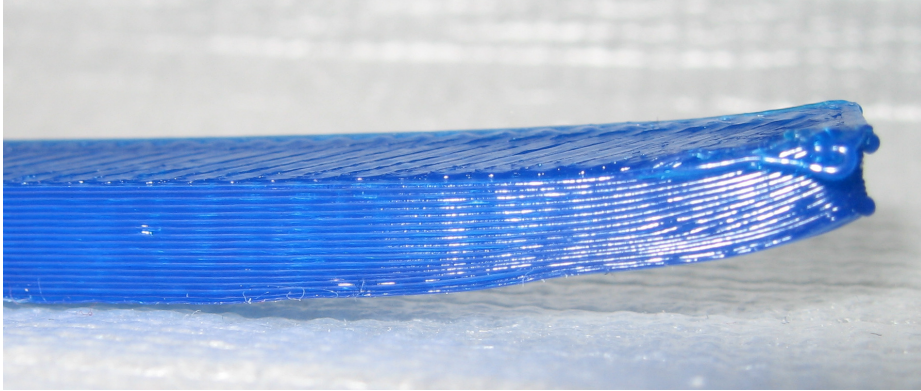


Figure 6.1: Commonly observed deformation in the first layers of 3-D printed PLA.

This study presents theory and experiments to investigate the mechanisms behind the deformation in PLA during FFF. First, existing models and their deficiencies in producing a remedy for deformation in 3-D printing PLA are reviewed. A 1-dimensional model is expanded to use a physics based temperature gradient, and it is verified using a measured temperature gradient in a printed part. The changes in size and crystallinity before and after annealing are measured to explain substrate release several hours into the print.

6.2 Background

6.2.1 Models of Deformation in FFF

Some very limited research has been done to prevent deformation in fused filament fabrication (and the subclass of fused deposition modeling, FDM). For plastic parts, a model was proposed to explain warpage [173] and tested on an FDM printer for number of layers, layer height and chamber temperature of an enclosed proprietary printer. This model uses a heated build chamber, so the temperature gradients were smaller. Because of this, it assumes that the gradient is a step function, where only the newly printed layer is not cooled to the chamber temperature yet. It thus simulates detaching a newly printed layer letting it cool to the chamber temperature and then straining that layer back to the original length, reattaching it and enforcing equilibrium conditions. Although this is non-physical it provides useful results particularly for the class of printer for which it was targeted. Based on this model,

an alternative method for toolpath creation has been proposed and demonstrated to limit warpage of the printed object [174]. This method limits the length of the traces by dividing the object into bricks. Their results show that this limits the warpage, as predicted by the model.

A second model was developed by Xinhua et al. This model also simplifies the temperature gradient in the object that is being printed as a step function, and assumes every layer to be deposited in zero time [175]. It is based on a model for deformation of a thin plate with a vertical load [176]. This does not correspond to the modeled scenario, and therefore it does not give relevant results. It is unclear how the final deformation is computed from the given formula. The system only varies in the vertical direction, and the model that was used predicts zero deformation for that case. Xinhua et al. then continue to determine optimal process parameters for thin plates based on experiments, but the model's predictions are not compared to the experimental results [175] and continuation of this model is not recommended.

Thus in this study, Wang et al.'s model is extended to a physics-based generalizable system of FFF without a heated chamber.

6.2.2 PLA and FFF

PLA is a thermoplastic polymer that is produced from natural sources such as corn. Its chemical formula is $(C_3H_4O_2)_n$. It is nontoxic and biodegradable, making it a popular choice for packaging and 3-D printing. Another feature that makes it good for 3-D printing is its limited shrinkage compared to other thermoplastics such as ABS.

Polymers can have varying degrees of crystallinity and PLA is no exception. Crystallinity of commercial PLA filament varies due in part to coloring agents [172]. Annealing causes PLA to crystallize, following standard Avrami crystallization kinetics [177]. Like other polymers, PLA can also be crystallized by stretching it while it is below the melting temperature, but above the glass transition temperature.

The PLA molecule is chiral, so there are two isomers of it, PLLA and PDLA. Pure forms of either of the isomers more easily crystallize, but the mixture has a higher melting point [178]. The density of amorphous PLLA is 1248 kg/m^3 , while for crystalline PLLA it is 1290 kg/m^3 , so there is a volume decrease of 3.3%. Crystallization of PLA takes place around nucleation sites in the form of spherulites: spherical structures of outward extending arms [177, 178]. The glass transition temperature and

melting temperature of PLA depend on the mixture of PLLA and PLDA and on the crystallinity. The glass transition temperature of semi-crystalline PLA is around 58°C and the melting temperature is between 130°C and 230°C [179].

During FFF 3-D printing, the PLA is first heated to a temperature near the melting temperature (T_m) but above the glass transition temperature (T_g). It is then extruded and deposited on the object that is being printed in the temperature range between T_m and T_g . During this time PLA can acquire a large deformation with less force and the strength is small [180]. The object cools as the print progresses and internal stress is accumulated [173]. The substrate may be heated, resulting in an even more complex temperature gradient in the object. Heating a chamber in which the 3-D printer is placed [181, 182] or a printer with an integrated heated build chamber such as the GolemD [183] or the Kühling & Kühling RepRap Industrial 3-D printer [184], on the other hand, keeps the differences smaller. Most proprietary 3-D printers have enclosed heated build chambers, which helps them limit the warpage.

The extrusion causes the material to decrystallize [185]. This effect is stronger at a lower temperature. The cooling leads to shrinkage, and because of the temperature gradient, this leads to internal stress and deformation. Finally, a heated substrate may anneal the lower layers of the object, and a heated chamber may anneal the entire object. This results in crystallization of the PLA. All of these processes will be probed to develop a full model.

6.3 Methods

6.3.1 Materials

All objects were printed from PLA that was extruded from resin (Natureworks 4043D 25kg polylactide resin) without additives [186]. Some properties of this material are summarized in Table 6.1.

6.3.2 3-D printing

The prints were made on a MOST RepRap 3-D printer [21], with a 0.5 mm nozzle. The substrate was regular soda-lime glass without heating. The substrate was cleaned

Table 6.1

Selected properties of the polymer that was used during the experiments.

Glass transition temperature (T_g)	$\approx 58^\circ\text{C}$
Melting temperature (T_m)	130–230°C
Linear coefficient of thermal expansion	$\approx 8.5 \cdot 10^{-6} \text{ K}^{-1}$
Diameter before extrusion	1.75 mm
Diameter of extrusion nozzle	0.5mm
Density	1248 – –1290 kg/m ³
Poisson’s ratio	0.36

with soap and distilled water and dried with compressed air. While application of glue to the substrate improves bed adhesion, it can interfere with x-ray diffraction (XRD) measurements and was avoided.

For all experiments, the 3-D printer was controlled using Franklin [1]. The objects that were measured with XRD were created with OpenSCAD [86] and sliced using Slic3r [187] with 100% infill. The infill was set to be lines that are parallel to the sides of the object, alternating between the long and the short direction. The toolpath for the temperature measurement was created using a custom Python script [188].

6.3.3 Measurements

6.3.3.1 Temperature

Temperature graphs of a thin vertical wall were made while printing using an infrared sensitive camera (FLIR e60). The recordings were stored as mp4 movie files by the camera, in false color with a scale bar on screen. Also included on screen was the temperature of the center of the image. Figure 6.2 is a frame from one recording.

The temperature of the pixels on the image was determined from the image. The red, green and blue channels of the pixels in the scale bar were plotted against their position in the scale bar. The plot was divided into regions where at least one component had a linear dependence on temperature. Those linear relations were manually fit in order to convert a pixel’s color value into a position on the scale. Because the upper and lower limits of the scale varied, and those variations were not updated immediately in the image, they also had to be determined from the image. The constant temperature background was used as a reference. The reported temperature of the

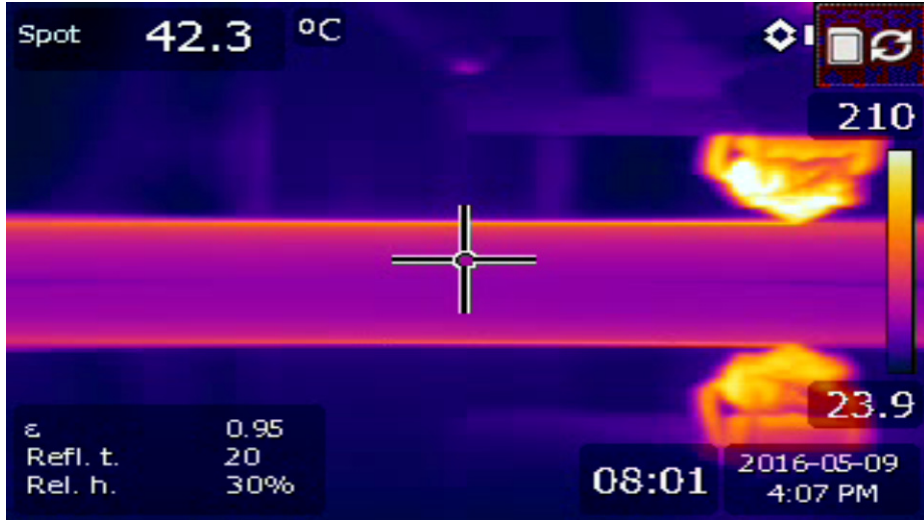


Figure 6.2: A representative infrared images that was used to measure the temperature of the printed object.

central pixel was updated immediately, and was used as the second reference. The two references were used to convert the position on the scale into a temperature.

The vertical temperature gradient of a thin vertical wall was examined while it was being printed. The toolpath started with a line next to the wall to prime the extruder. Then the nozzle retracted and moved to a parking position behind the wall. It then moved to the middle of the wall, unretracted while it moved to the left side, extruded a line of filament while moving to the right, then another line just behind it while going back to the left. Then it moved back to the center while retracting, followed by a move back to the parking position. The move to the center ensured that the wall was not pulled sideways. This was important because it was printed on bare glass, and would be pulled off the substrate otherwise. After this, the same procedure was repeated for the next layer, with left and right reversed. Layers were added similarly until the top of the object was reached. The images that were used for analysis were always every second layer, with the extruder in the same position and moving in the same direction.

6.3.3.2 Temperature Model

To enable extension and improvement of Wang et al.'s work to a more generalizable FFF 3-D printing scenario a physics-based model for the temperature gradient in the printed object was derived. The existing model uses a step function for the

temperature. This is a simplification that is only valid if the environment temperature is relatively high. The improvement in this work was to replace the step function with a physics-based temperature gradient that was fit to a set of measured temperatures.

The change in temperature with time has three components. Conductivity is proportional to the second derivative of the temperature to the position, convection is proportional to the temperature difference between the object and the air, and radiation is proportional to the fourth power of the absolute temperature of the object. There is also incoming radiation, which is proportional to the fourth power of the absolute environment temperature.

The model assumes symmetry in both horizontal directions, so z is the only spatial variable of interest:

$$\frac{\partial T}{\partial t} = C_0 \frac{\partial^2 T}{\partial z^2} - C_1 \Delta T - C_2 \Delta(T^4) \quad (6.1)$$

In this equation, C_0 and C_1 are proportionality constants for conduction and convection respectively, and C_2 is the Stefan-Boltzmann constant multiplied with the object surface.

This equation does not have an analytical solution. Therefore, it was simplified to ignore radiation. This simplification is justified for temperatures close to the environment temperature, where conduction and convection are much more important factors for heat transport than radiation. For FFF 3-D printing, this is applicable. For an object of 200°C in an environment of 20°C, radiation accounts for a heat loss of only 10 $\mu\text{W}/\text{m}^2$.

3-D printing is modeled as a continuous process. The top of the object grows with time at a speed v . The temperature difference between the top boundary and the air is always T_0 .

With this simplification and boundary condition, the solution to the equation is:

$$\Delta T(z, t) = T_0 e^{\frac{a}{v}z - at} \quad (6.2)$$

In this solution, a is a function of C_0 , C_1 and v . Because C_1 depends on many properties, including ambient temperature, air flow and humidity, no attempt is made

to predict its value, and instead a is fit to the measurement. The only time of interest is at the end of the print, so $t = \frac{h}{v}$.

The temperature gradient above (Equation 6.2) was applied in Wang's model instead of the step function that was used there. The other two equations remain identical:

$$\int_0^h (-E\alpha\Delta T + \sigma' + \frac{Ez}{R})dz = 0 \quad (6.3)$$

$$\int_0^h (-E\alpha\Delta T + \sigma' + \frac{Ez}{R})zdz = 0 \quad (6.4)$$

In these equations, E is the elastic modulus which is 1.03 GPa for amorphous PLA [177], and α is the linear shrinkage coefficient, which is $8.5 \cdot 10^{-6} \text{ K}^{-1}$ [165]. σ' is a combination of several stress components, which is simplified in the original model by treating it as a scalar. This simplification is used in this improved model as well.

Equations 6.2, 6.3 and 6.4 can be solved for the curvature $k = \frac{1}{R}$:

$$k(h) = \frac{6\alpha T_0}{h^2} \left(2 + \frac{2v^2 - a^2h}{ahv} \left(e^{-\frac{h}{v}a} - 1 \right) \right) \quad (6.5)$$

To verify this model, a wall was printed as in the temperature measurement, and it was broken off the substrate immediately after completion of the print, and left to cool to room temperature normally. It should be noted that there is a difference between this method and what is modeled. The substrate is not part of the model, and therefore the predictions for layers closer to the substrate are unreliable, because the thermal conductivity and heat capacity of the substrate differ from a continuing wall of PLA. For high objects, this introduces a small error, while for low objects, the error will be larger, because the layers of those objects are on average closer to the substrate.

The resulting warped objects were scanned at 600 dpi on a flat bed scanner and

the scans were analyzed using Gimp [189] using its measurement tool to measure distances and angles in the image. The height of the object (h) was verified to be what it was printed as, and the distance from the line between the two tips of the object to the lowest point on the top layer, in the middle (δ), was measured by first rotating the image to make this line vertical, and then using the rectangular select tool to measure the horizontal distance. This is the amount of warp. From this and the length of the object, the radius of curvature (R) was computed and compared to the predicted value. A diagram with the variables is shown in Figure 6.3.

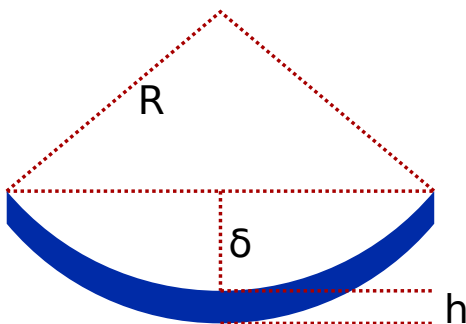


Figure 6.3: Diagram showing the meaning of the variables in an exaggerated example of warping.

6.3.3.3 Crystallinity

Rectangular blocks of $10 \text{ cm} \times 2.8 \text{ cm} \times 1 \text{ cm}$ were printed on bare glass as shown in Figure 6.4 and annealed at 50°C between 0 and 4 hours. This temperature is just below the glass transition temperature of approximately 58°C and was chosen as representative for a typical heated substrate during a print. The size before and after annealing was measured, and crystallinity was measured using XRD with a Scintag XDS 2000 theta-theta powder diffractometer with a copper X-ray tube, 1 mm and 2 mm beam slits and 0.3 mm and 0.5 mm receiving slits and a graphite monochromator. The printed block was placed in the sample holder with the smooth bottom layer facing the instrument. The X-ray source and the detector were moved simultaneously and the intensity of the reflection was measured at each angle. Crystallinity was determined by dividing the area of the crystalline peaks by the total area of the measurement.

The parts are expected to crystallize during annealing, and shrink as a result of that. Because no crystallinity was observed even as a result of 4 hours of annealing, the blocks were annealed again for the same time as before, but now at 90°C . The size

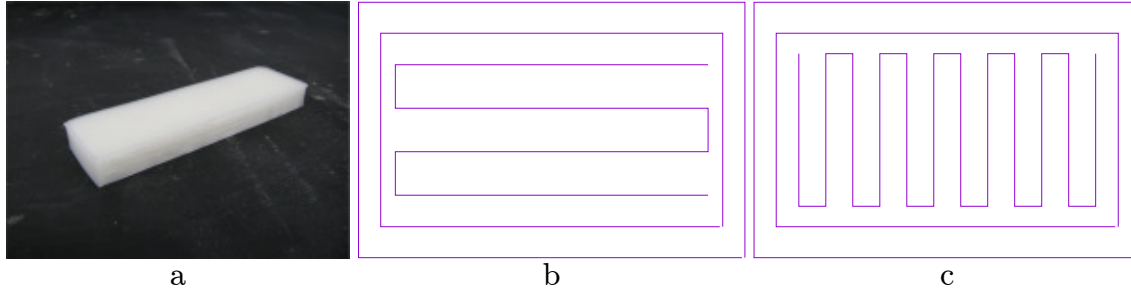


Figure 6.4: Samples that were used for the XRD measurements. a) Photograph of the sample, dimensions are $10\text{ cm} \times 2.8\text{ cm} \times 1\text{ cm}$. b) Diagram of the print path for odd layers. c) Diagram of the print path for even layers. The number of lines in the diagrams of the print paths has been reduced for clarity.

and crystallinity were determined again after this.

In order to better understand the deformation during annealing, the object that was not annealed and the one that was annealed for 1 hour at both temperatures were frozen with liquid nitrogen and broken with a pair of pliers while holding them in a vise. The parts were left to warm up to room temperature and for both samples the interface was inspected with an FEI XL-40 environmental scanning electron microscope (ESEM), using the backscatter electron detector, 10 kV accelerating voltage and a working distance of 10 mm.

6.4 Results

An example frame that was used for one curve is shown in Figure 6.2 with the hotend moving to the right. As can be seen in Figure 6.2, the clean glass substrate is highly reflective at the large angle and these reflections were not analyzed. It is also clear that the hottest PLA has just left the nozzle and the coldest PLA is on the bottom layer on the return side, which has had the longest to cool. There is a clear temperature gradient from the substrate surface to the print surface. The emissivity of the PLA is high at 0.95. This is good for the precision of the measurement. Figure 6.5 shows the gradient of the temperature in the vertical wall in steps of two layers. Every curve had the print head in the same position and moving in the same direction. The curves become longer as the print becomes taller. The maximum intensity is the most recently printed layer. The thermal model was fit to the data and is shown in Figure 6.5 as smooth lines. The lines deviate significantly from the measured

temperatures. The model assumes a continuous growth of the top, while in reality it is built in layers. The top is kept at a constant temperature in the model. The average temperature of the top layer in reality depends on the cooling rate. Insofar as the model is correct, the height at which the average is constant may not be exactly at the top of the print. This could have been included in the model as an extra parameter. However, doing this led to the model becoming unstable, due to too many parameters.

The data drops more steeply than the model at the higher temperatures. This causes the base of the fit to be higher than the data. This means that the model predicts the curves to be closer to the environment temperature than they actually are, while the environment temperature is fit to a higher value than it actually is. The fit found an environment temperature of 45°C while it was around 20°C in reality, and an average top layer temperature of 74°C. The actual value of the top layer temperature can be seen in the graph. The value of a , the parameter that combines several constants, was fit to 0.012 s⁻¹. The high environment temperature is explained by the design of the 3-D printer, which blows air past the extruder and onto the object. This air is warmed up by the extruder, and therefore the environmental air of the object during printing is higher than room temperature.

Before annealing, but also after annealing the objects at 50°C, no crystalline peaks were visible on the XRD spectra. The XRD spectrum for the sample that was annealed for 4 hours at 50°C, and the spectrum for the sample that was annealed for two hours at 50°C and then 2 hours at 90°C are shown in Figure 6.6. As can be seen, the sample annealed at 50°C is completely amorphous, while the sample annealed at 90°C clearly shows crystallinity. To determine the crystallinity, it was first determined that three broad Pearson 7 peaks fit the amorphous spectrum well. Then in the spectra with crystalline peaks, the amorphous background was fit to three broad peaks as well. The total area under the crystalline peaks was divided by the total area of all peaks to determine the crystallinity of the sample.

After annealing for 1 hour at 90°C, the crystallinity was almost 20%, and further annealing did not make it go up much further (Figure 6.7).

The size of the objects is shown in Figure 6.8. The absence of any change for the 50°C annealing can clearly be seen, as can the effect of a smaller horizontal size after annealing at 90°C. However, the height of the objects increased.

The height increase is also visible in the images taken with the ESEM, which is shown in Figure 6.9. It is clear from the image that the calibration of the 3-D printer was not optimal, because while it was instructed to give the object solid infill, there are

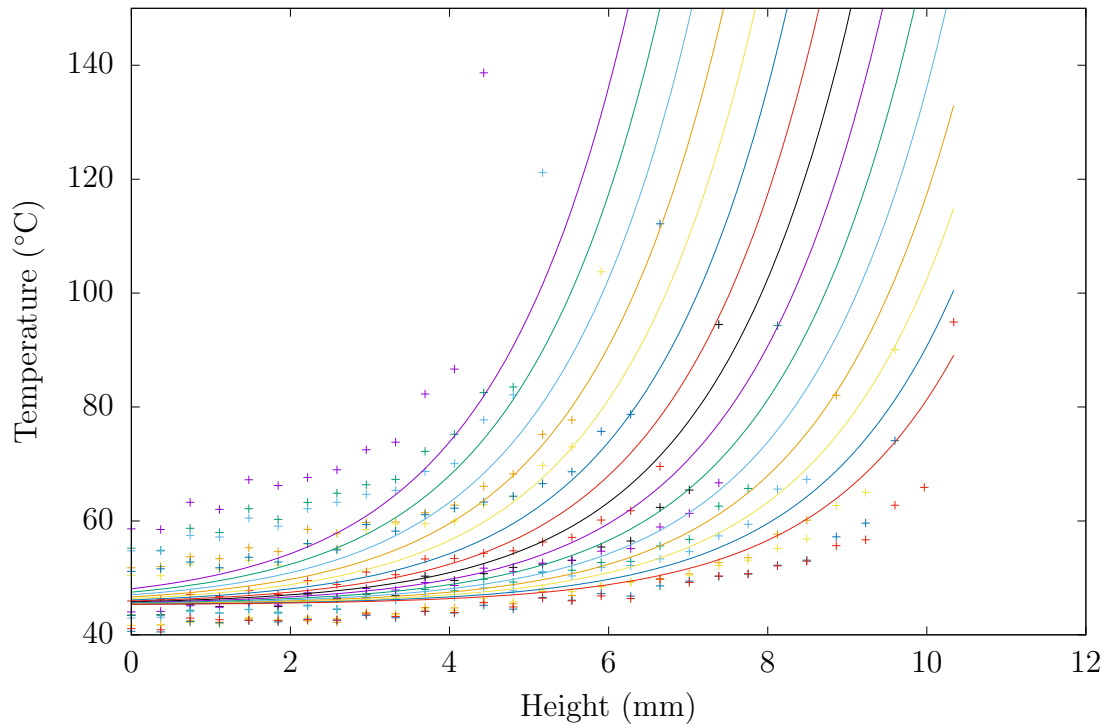


Figure 6.5: Vertical gradient of the temperature at several points in time. The curves are the fit of the thermal model to the data.

large air spaces present. The lines are also grouped in pairs, which suggests hysteresis in the machine that causes the lines that are printed while the nozzle is moving one way to be slightly offset from the lines that are printed while the nozzle is going the other direction.

The measurements on the warped vertical wall are shown in Figure 6.10. The curve is the prediction from the model, purely based on the thermal data. It is very close to the actual values given that the temperature fit had a sizable error and simplifications were made to the physics. The points at larger heights are expected to be closer to the curve, because the substrate is not part of the model and its effect is smaller at greater thicknesses. The method also expected the part to start warping while the head was on the last layer in the position as shown in Figure 6.2, but in practice it took a few seconds to break it off the substrate, so it had cooled a little more than what the model expected. Because of this, the values are expected to be slightly lower than what the model predicts.

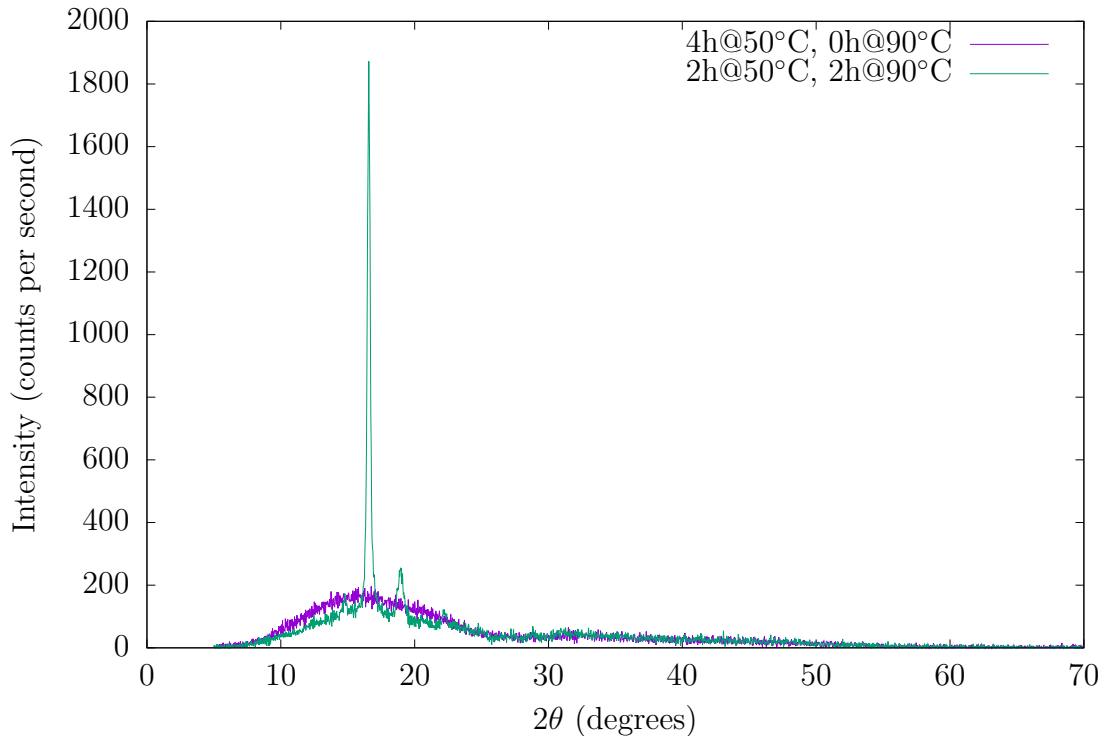


Figure 6.6: XRD spectra for the sample that was only annealed at 50°C, and the sample that was first annealed for two hours at 50°C, then for two hours at 90°C.

6.5 Discussion

The thermal model presented here is based on physics and a real world temperature gradient. It is therefore more realistic than what was used by Wang et al. [173]. The model successfully predicts warpage of amorphous objects. The expectation is that it also works for other plastics, but that remains to be tested. The behavior of the curvature is correctly predicted, but the magnitude is too high, especially for lower print layers. This is likely caused by the simplifications in the model, such as ignoring the effects of the substrate. This has a larger effect on lower prints, so the larger error in the prediction was expected.

Some experiments were also done to show the effect of annealing on printed objects. This is especially relevant for printers with a heated substrate or chamber, because those printers are annealing the objects during the print. It was found that at a temperature of 50°C, which is just below the glass transition temperature of 58°C,

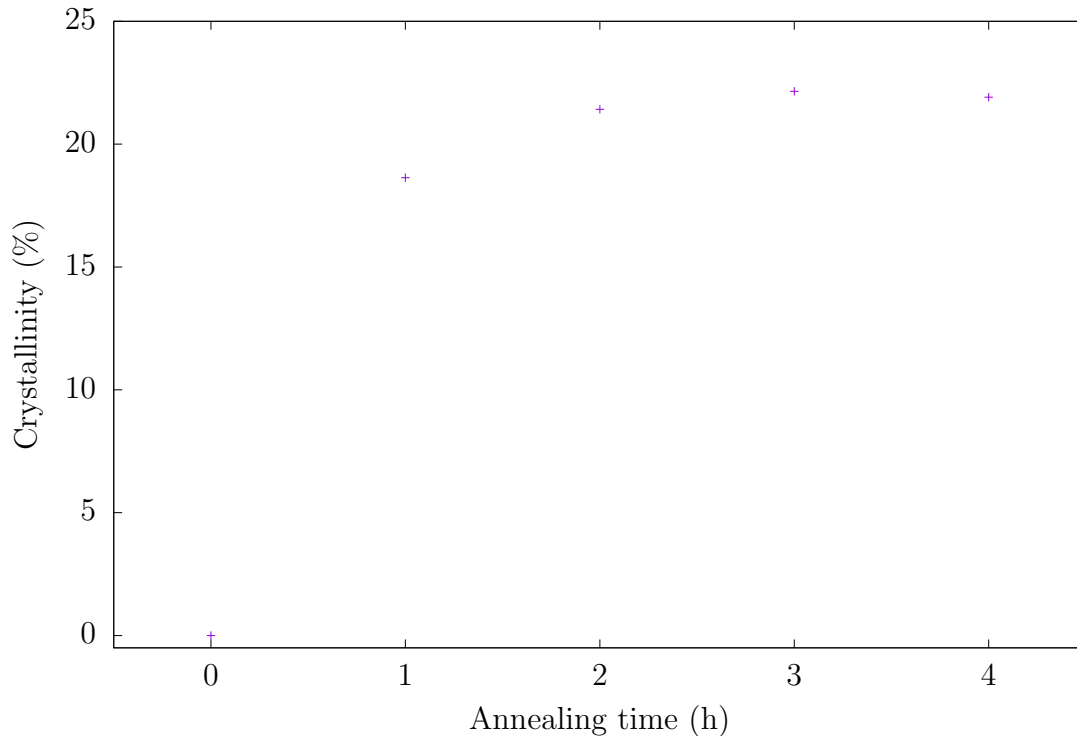


Figure 6.7: Crystallinity as a function of annealing time when annealing at 90°C.

the objects neither shrink nor crystallize, while at 90°C, they rapidly crystallize and shrink considerably. In order to draw conclusions about this effect, it must first be studied as a function of infill. Colored PLA already has some crystallinity as shipped [172]. Further study is required to see if this means that it shrinks less, or that it more readily crystallizes and reaches higher crystallinity, thus shrinking more. The crystallization explains why long prints can release from the substrate several hours into the print. This suggests that keeping the temperature of the substrate below 50°C would prevent this problem with natural PLA.

An unexpected effect of annealing was that the height of the objects increased. While there was a 5% size decrease in both horizontal dimensions, there was an 8% increase in the vertical dimension, so the average volume decrease per dimension is only 0.8%. Crystallization is expected to cause a volume decrease of 3.3%, so at 22% crystallinity that is only a 0.72% volume decrease, or 0.24% per dimension. The other 0.6% of shrinkage per dimension may be caused by relaxation of residual stress. An 8% increase in z means a shrinkage in x and y of 3% with Poisson's ratio $\nu = 0.36$. This is less than the observed 5% of shrinkage.

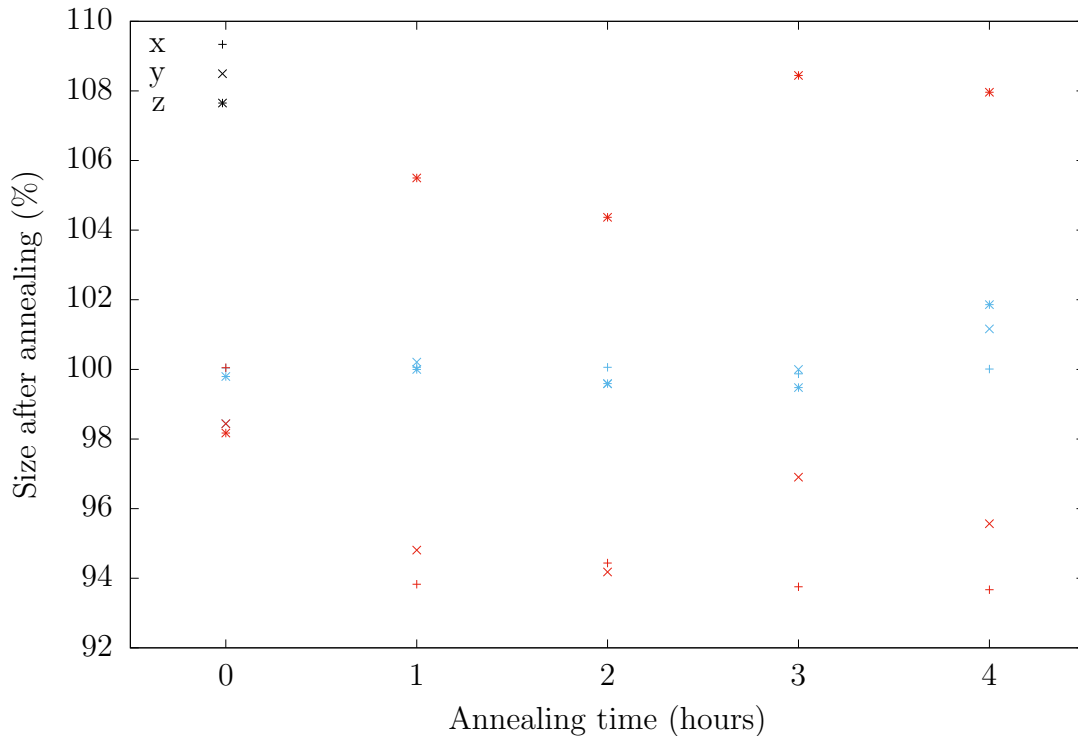


Figure 6.8: Changes in size after annealing at 50°C and 90°C. At 50°C (blue markers) there is no change in size, while at 90°C (red markers) the objects have shrunk horizontally, but grown vertically.

An alternative approach is to consider the shrinkage that would have occurred if the object were not held by the substrate. This is the shrinkage that would be expected as a result of complete stress relaxation, assuming that no relaxation has taken place while the object was constrained. The linear shrinkage coefficient is $8.5 \cdot 10^{-6} \text{ K}^{-1}$, so for a temperature drop from 210°C at extrusion to 25°C, a shrinkage of 1.6% would be expected. This is far below the 3.0% that was derived above, so the increased height cannot be explained solely by horizontal stress relaxation.

The ESEM images show an additional effect, which can be responsible for a part of the height increase. Before annealing, the cross section of the printed filament is flattened, and after annealing it is more circular, as can be seen in Figure 6.9. The ratio between width and height, measured from the images, is 0.82 for the sample that was annealed at 90°C and 0.74 for the sample that was not (both with a standard deviation of 0.08). This may be caused by the material decreasing its surface energy.

The imaged parts show different deformation than the complete sample: 10.1% horizontal shrinkage, and 3.4% vertical expansion. The ratio in density can be determined



Figure 6.9: The surface of the fracture interface of 1 hour annealed (top) and unannealed (bottom) samples, taken with an Environmental Electron Scanning Microscope (ESEM).

from these numbers, by assuming that the view in the x and y directions is the same. Measuring the lengths of equal parts in both images and multiplying the horizontal length squared with the vertical length gives a volume. This volume has equal mass in both images, so the ratio of densities can be found by dividing them. This gives a

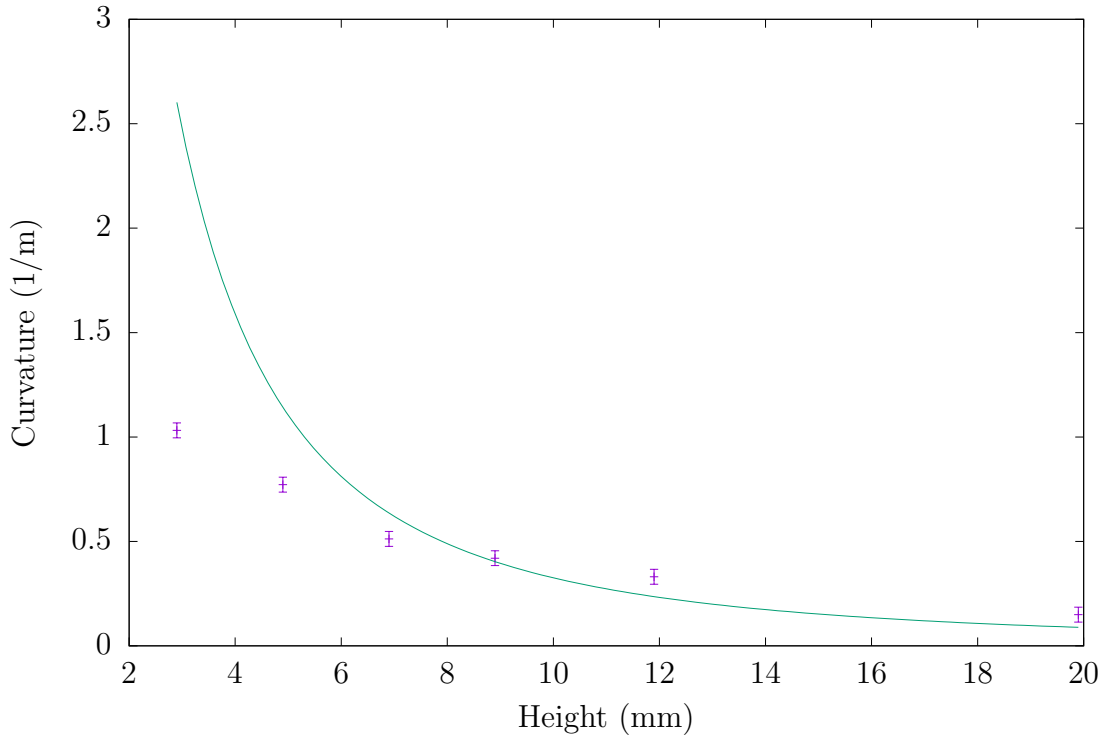


Figure 6.10: Measured warp and model predictions (Equation 6.5). The curve is the model’s prediction with the variables from the fit to the thermal graph.

ratio of 83.4%. According to the theory, amorphous PLLA is 1248 kg/m^3 , while crystalline PLLA is 1290 kg/m^3 . This is a ratio of 96.7%, and because the crystallinity is around 20%, not 100%, the ratio due to crystallinity in these samples is 99.3%.

Analyzing the images in Figure 6.9 using Gimp, it was determined that the air content before annealing was 21.3%, while after annealing it was 21.5%. Given the magnitude of the changes in size, this can be safely ignored.

These results can also be used to better guide slicing routines for large objects printed in PLA or other polymers that build up internal stress. In Wang’s article [173] a combination contour raster is chosen, which takes advantage of the good surface finish and dimensional accuracy of a contour on the outer layers, and then the high speed deposition of a raster on the interior. Such a method of separating fill and exterior is common in all open source slicing programs (e.g. Cura and Slic3r). Guerrero had proposed an alternative method for toolpath creation that has demonstrated less warpage of the printed object [174]. It involves breaking up long runs into bricks. With the same curvature, this limits deformation by shortening the length of the curving pieces, but this comes at the cost of loss in strength. In addition, the actual

toolpath that was used for bricking was not divulged. Future work is needed to determine the optimal size of bricks, the shape (square or hexagonal), the stacking orientation, the toolpath within a brick (e.g. raster vs contour or some combination and the level of infill) and the printing temperature.

Although this study successfully explained how the deformation of PLA is caused by the temperature gradient that builds up during the print, it did not investigate how to stop it. However, the results of this study indicate a potential solution. It involves using the model developed here, improving upon it by excluding the simplifications, and then using it to predict the exact deformation of a brick building block. Then an object must be sliced into bricks with space for mortar of the size needed to fill the expected deformation on a sealing pass. The mortar must be printed after the layer has cooled and shrunk, and would ideally be determined by the nozzle diameter of the FFF machine. Finally, the mortar tool path would need to be optimized to minimize single toolpath lengths while binding all the bricks into a single unit. Future work is necessary to test this new method of slicing to prevent fatal deformation of large-scale FFF, and how it affects the mechanical properties of the printed objects.

6.6 Conclusions

The deformation model presented by Wang et al. for chamber based FDM was generalized for FFF 3-D printing with a room temperature environment and expanded to use a physics based temperature gradient. The thermal equation was calibrated using thermal measurements and validated by measuring curvature in printed objects. The results showed that this makes the model usable for lower ambient temperatures. It is not very accurate for the first layers, but after about 9 mm it predicts the deformation well. Additionally, the effect of annealing was examined. It was found that at a temperature of 50°C, no shrinkage or crystallization takes place, but at 90°C the PLA rapidly crystallizes to around 20% crystallinity. This indicates that heated bed temperatures should be maintained at 50°C or lower to avoid delamination of the print from the substrate with PLA. At 90°C, the annealing is accompanied by a 5% size decrease in both horizontal dimensions, but an 8% increase in the vertical dimension. Thus, the volume decreased by 3%, indicating some potential slicing measures that could reduce deformation of a printed object.

Chapter 7

Future Work

This dissertation describes a few steps on the way of making science more accessible for everyone through the use of 3-D printers. It is highly interdisciplinary. The computer science development in Chapter 2 gives greater control over materials processing equipment, such as 3-D printers, and materials characterization equipment, such as the 3-D microscope described in Chapter 5. The work on computer science and electrical engineering in Chapter 3 allows for low cost materials characterization in remote areas. Similarly, the work on mechanical and electrical engineering in Chapter 4 is helpful to medical doctors, chemists, physicists, biologists and materials scientists for process control of their work. The microscope described in Chapter 5 allows scientists from all disciplines to do automated 3-D microscopy for any purpose, including materials characterization. Finally, Chapter 6 presents a physics based model which can be used by materials scientists to better explain the structure and properties of poly lactic acid as it is being printed.

Much more can be done in regards to making science more accessible. In this final chapter, ideas for future work will be provided.

7.1 3-D Printer Improvements

First of all, several improvements to open source 3-D printers would be useful for scientific applications. Delta printers are not easy to calibrate. A machine such as the Athena[135] allows mounting a microscope, as described in Chapter 5. This feature should be used in combination with focus detection to determine the position of the

build platform in many locations. Then those positions can be used to automatically calibrate the machine.

As described in Chapter 6, a heated build chamber would be a valuable feature to prevent deformation.

Once the deformation is under control, the option of printing larger parts becomes available. For this, a large scale printer would be needed, and it must be set up to deposit more material per second, so that printing completes in a reasonable time.

7.2 Software Improvements

One major benefit of using open source hardware is that it allows the user to adapt the device to their needs. This is even more powerful when combined with open communication standards. There is currently no universal standard for automated sensors. It would be very useful if such a standard were to be developed, so that programs that do data acquisition and transfer can handle sensors that were unknown to the developer of the software.

Related to this is the problem of ownership. For repositories that hold a lot of information, such as Thingiverse [190] or Github [191], the owners of those services can do significant damage to the community by taking their service offline. In a more subtle way, they can damage the community by modifying the presentation of the information to benefit their interests instead of those of the users. The best protection against both those dangers is to decentralize the services, so they become webs of servers hosting small parts of the information each. This presents significant challenges, especially in terms of searching through the information. File sharing services have solved some of those issues. It would be good if that knowledge was used to set up a web of scientific information that is not controlled by any single entity.

7.3 Education

As part of my work here, I have helped to run workshops to build 3-D printers with middle and high school teachers. These workshops are extremely good for society,

because they give access to open source hardware to the students. Not only that, the printers are devices that allow them to build their own open source hardware, so they learn about the possibility of designing their own lives. It would be good if more such workshops would be given. And because teachers are always busy and do not have much time to develop new material, in addition it would be good if more educational programs would be developed to make use of 3-D printers.

Another way to make 3-D printers more useful to schools, is by going through the entire curriculum of a discipline, such as physics or biology, identify every tool that could be printed to help with the subject, and make designs for those tools, similar to what was done by Zhang et al. for optics equipment [65]. That would result in an open source library which would lead to big cost reductions for schools.

7.4 Deformation

The improved deformation model that was presented in Chapter 6 should be used as a basis for preventing or compensating for deformation. Especially for large printers, it would make the machine easier to build and less costly if it would not have a heated build platform or build chamber. Therefore, temperature should be controlled through delays and possibly cooling fans. That limits the control to cooling, so the slicing strategy must account for that.

One option that could be explored would be to use a bricked raft on which a normally sliced object is built. This would allow some flexibility compared to attaching the object directly to the substrate, which could help the object to stay attached to the raft.

References

- [1] B. Wijnen, G. C. Anzalone, A. S. Haselhuhn, P. Sanders, and J. M. Pearce, “Free and open-source control software for 3-D motion and processing,” *Journal of Open Research Software*, vol. 4, no. 1, 2016.
- [2] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer, “RepRap—the replicating rapid prototyper,” *Robotica*, vol. 29, no. 01, pp. 177–191, 2011.
- [3] E. Sells, Z. Smith, S. Bailard, A. Bowyer, and V. Olliver, “RepRap: the replicating rapid prototyper: maximizing customizability by breeding the means of production,” *Handbook of Research in Mass Customization and Personalization, Forthcoming*, 2010.
- [4] A. Bowyer, “3D printing and humanity’s first imperfect replicator,” *3D printing and additive manufacturing*, vol. 1, no. 1, pp. 4–5, 2014.
- [5] “Pronterface, pronsole, and printcore — pure python 3d printing host software.” <https://github.com/kliment/Printrun>.
- [6] “Cura download page.” <http://software.ultimaker.com>.
- [7] “Octoprint 3-D printer controller.” <http://octoprint.org>.
- [8] “Pacemaker, a host-client-system to control 3d printers, CNC milling or laser cutters.” <https://github.com/JustAnother1/Pacemaker>.
- [9] T. Wohlers, *Additive Manufacturing and 3D Printing State of the Industry: Annual Worldwide Progress Report, Wohler’s report 2013*. Wohlers Associates, 2013.
- [10] J. M. Pearce *et al.*, “Building research equipment with free, open-source hardware,” *Science*, vol. 337, no. 6100, pp. 1303–1304, 2012.
- [11] J. M. Pearce, “Open-source nanotechnology: Solutions to a modern intellectual property tragedy,” *Nano Today*, vol. 8, no. 4, pp. 339–341, 2013.

- [12] C. L. Ventola, “Medical applications for 3D printing: current and projected uses,” *PT*, vol. 39, no. 10, pp. 704–711, 2014.
- [13] B. Wijnen, E. J. Hunt, G. C. Anzalone, and J. M. Pearce, “Open-source syringe pump library,” *PloS one*, vol. 9, no. 9, p. e107216, 2014.
- [14] P. Drescher, S. Spath, and H. Seitz, “Fabrication of biodegradable, porous scaffolds using a low-cost 3D printer,” *International Journal of Rapid Manufacturing*, vol. 4, no. 2-4, pp. 140–147, 2014.
- [15] D. L. King and A. Babasola, “Mobile open-source solar-powered 3-D printers for distributed manufacturing in off-grid communities,” *Challenges in Sustainability*, vol. 2, no. 1, p. 18, 2014.
- [16] J. M. Pearce, C. M. Blair, K. J. Laciak, R. Andrews, A. Nosrat, and I. Zelenika-Zovko, “3-D printing of open source appropriate technologies for self-directed sustainable development,” *Journal of Sustainable Development*, vol. 3, no. 4, p. 17, 2010.
- [17] T. Birtchnell and W. Hoyle, *3D printing for development in the global south: The 3D4D challenge*. Palgrave Macmillan, 2014.
- [18] B. Wijnen, “Franklin, 3-D printer and CNC controller.” <https://github.com/mtu-most/franklin>.
- [19] “Arduino prototyping platform.” <http://www.arduino.cc>.
- [20] B. T. Wittbrodt, A. Glover, J. Laureto, G. Anzalone, D. Oppliger, J. Irwin, and J. M. Pearce, “Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers,” *Mechatronics*, vol. 23, no. 6, pp. 713–726, 2013.
- [21] J. L. Irwin, J. Pearce, G. Anzalone, and D. Oppliger, “The RepRap 3-D printer revolution in STEM education,” in *121st ASEE Annual Conference & Exposition*, 2014.
- [22] G. C. Anzalone, B. Wijnen, and J. M. Pearce, “Multi-material additive and subtractive prosumer digital fabrication with a free and open-source convertible delta RepRap 3-D printer,” *Rapid Prototyping Journal*, vol. 21, no. 5, pp. 506–519, 2015.
- [23] G. C. Anzalone, C. Zhang, B. Wijnen, P. G. Sanders, and J. M. Pearce, “A low-cost open-source metal 3-D printer,” *Access, IEEE*, vol. 1, pp. 803–810, 2013.

- [24] A. S. Haselhuhn, E. J. Gooding, A. G. Glover, G. C. Anzalone, B. Wijnen, P. G. Sanders, and J. M. Pearce, “Substrate release mechanisms for gas metal arc weld 3D aluminum metal printing,” *3D Printing and Additive Manufacturing*, vol. 1, no. 4, pp. 204–209, 2014.
- [25] R. W. Hamming, “Error detecting and error correcting codes,” *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [26] “Valgrind instrumentation framework for building dynamic analysis tools.” <http://valgrind.org>.
- [27] “Firebug web development tool.” <http://getfirebug.com>.
- [28] “Debian — the universal operating system.” <https://www.debian.org>.
- [29] “Raspberry pi.” <http://www.raspberrypi.org>.
- [30] “BeagleBone Black.” <http://beagleboard.org>.
- [31] “Debian backports.” <http://backports.debian.org>.
- [32] “Python-fhs.” <https://github.com/wijnen/python-fhs>.
- [33] “Filesystem hierarchy standard.” <http://www.linuxfoundation.org/tags/filesystem-hierarchy-standard>.
- [34] “Python-network.” <https://github.com/wijnen/python-network>.
- [35] “Python-websocketd.” <https://github.com/wijnen/python-websocketd>.
- [36] B. Wijnen, G. Anzalone, and J. M. Pearce, “Open-source mobile water quality testing platform,” *Journal of Water Sanitation and Hygiene for Development*, vol. 4, no. 3, pp. 532–537, 2014.
- [37] E. Johansson, T. Wardlaw, N. Binkin, C. Brocklehurst, and T. Dooley, “Diarrhoea: Why children are still dying and what can be done,” 2009.
- [38] “WHO 2013 diarrhoeal disease, fact sheet no. 330.” <http://www.who.int/mediacentre/factsheets/fs330/en>, 2013. (Accessed July 17, 2013).
- [39] S. A. Esrey, J. B. Potash, L. Roberts, and C. Shiff, “Effects of improved water supply and sanitation on ascariasis, diarrhoea, dracunculiasis, hookworm infection, schistosomiasis, and trachoma,” *Bulletin of the World Health organization*, vol. 69, no. 5, p. 609, 1991.
- [40] S. Cairncross, C. Hunt, S. Boisson, K. Bostoen, V. Curtis, I. C. Fung, and W.-P. Schmidt, “Water, sanitation and hygiene for the prevention of diarrhoea,” *International journal of Epidemiology*, vol. 39, no. suppl 1, pp. i193–i205, 2010.

- [41] P. Grau, “Low cost wastewater treatment,” *Water Science and Technology*, vol. 33, no. 8, pp. 39–46, 1996.
- [42] M. Khan, K. Yamamoto, and M. Ahmed, “A low cost technique of arsenic removal from drinking water by coagulation using ferric chloride salt and alum,” *Water Science and Technology: Water Supply*, vol. 2, no. 2, pp. 281–288, 2002.
- [43] C. Newton and J. Wilson, “Recirculating gravel filters: high-performance treatment at low cost for two small communities,” *Water Science & Technology*, vol. 58, no. 6, 2008.
- [44] S. A. Brownell, A. R. Chakrabarti, F. M. Kaser, L. G. Connelly, R. L. Peletz, F. Reygadas, M. J. Lang, D. M. Kammen, and K. L. Nelson, “Assessment of a low-cost, point-of-use, ultraviolet water disinfection technology,” *Journal of water and health*, vol. 6, no. 1, pp. 53–65, 2008.
- [45] J. Brown and M. D. Sobsey, “Microbiological effectiveness of locally produced ceramic filters for drinking water treatment in cambodia,” *Journal of water and health*, vol. 8, no. 1, pp. 1–10, 2010.
- [46] F. Hashmi and J. M. Pearce, “Viability of small-scale arsenic-contaminated-water purification technologies for sustainable development in pakistan,” *Sustainable Development*, vol. 19, no. 4, pp. 223–234, 2011.
- [47] Q. Mahmood, S. A. Baig, B. Nawab, M. N. Shafqat, A. Pervez, and B. S. Zeb, “Development of low cost household drinking water treatment system for the earthquake affected communities in northern pakistan,” *Desalination*, vol. 273, no. 2, pp. 316–320, 2011.
- [48] J. Simonis and A. Basson, “Manufacture of a low-cost ceramic microporous filter for the elimination of microorganisms causing common diseases,” *Journal of Water Sanitation and Hygiene for Development*, vol. 3, no. 1, pp. 42–50, 2013.
- [49] A. Mustafa, M. Scholz, S. Khan, and A. Ghaffar, “Application of solar disinfection for treatment of contaminated public water supply in a developing country: field observations,” *Journal of water and health*, vol. 11, no. 1, pp. 135–145, 2013.
- [50] R. Buamah, R. A. Mensah, and A. Salifu, “Adsorption of fluoride from aqueous solution using low cost adsorbent,” *Water Science and Technology: Water Supply*, vol. 13, no. 2, pp. 238–248, 2013.
- [51] R. M. Conroy, M. Elmore-Meegan, T. Joyce, K. G. McGuigan, and J. Barnes, “Solar disinfection of drinking water and diarrhoea in maasai children: a controlled field trial,” *The Lancet*, vol. 348, no. 9043, pp. 1695–1697, 1996.

- [52] R. M. Conroy, M. E. Meegan, T. Joyce, K. McGuigan, and J. Barnes, “Solar disinfection of water reduces diarrhoeal disease: an update,” *Archives of disease in childhood*, vol. 81, no. 4, pp. 337–338, 1999.
- [53] R. M. Conroy, M. Meegan, T. Joyce, K. McGuigan, and J. Barnes, “Solar disinfection of drinking water protects against cholera in children under 6 years of age,” *Archives of disease in childhood*, vol. 85, no. 4, pp. 293–295, 2001.
- [54] R. Meierhofer and M. Wegelin, “Solar water disinfection: a guide for the application of SODIS, SANDEC, EAWAG.” http://www.sodis.ch/methode/anwendung/ausbildungsmaterial/dokumente_material/manual_e.pdf, 2002. (Accessed July 17, 2013).
- [55] “Household water treatment, solar disinfection 2012.” <http://www.cdc.gov/safewater/PDF/Solar2011final.pdf>, 2012. (Accessed August 5, 2013).
- [56] J. Graf, S. Z. Togouet, N. Kemka, D. Niyitegeka, R. Meierhofer, and J. G. Pieboji, “Health gains from solar water disinfection (SODIS): evaluation of a water quality intervention in yaounde, cameroon,” *Journal of Water and Health*, vol. 8, no. 4, pp. 779–796, 2010.
- [57] B. Dawney and J. M. Pearce, “Optimizing the solar water disinfection (SODIS) method by decreasing turbidity with NaCl,” *Journal of Water Sanitation and Hygiene for Development*, vol. 2, no. 2, pp. 87–94, 2012.
- [58] B. Dawney, C. Cheng, R. Winkler, and J. M. Pearce, “Evaluating the geographic viability of the solar water disinfection (SODIS) method by decreasing turbidity with NaCl: A case study of south sudan,” *Applied Clay Science*, vol. 99, pp. 194–200, 2014.
- [59] J. M. Pearce, *Open-source lab: how to build your own hardware and reduce research costs*. Newnes, 2014.
- [60] F. P. Deek and J. A. McHugh, *Open source: Technology and policy*. Cambridge University Press, 2007.
- [61] J. Buitenhuis and J. Pearce, “Open design-based strategies to enhance appropriate technology development,” in *National Collegiate Inventors and Innovators Alliance. Proceedings of the... Annual Conference*, p. 1, National Collegiate Inventors & Innovators Alliance, 2010.
- [62] E. Stokstad, “Open-source ecology takes root across the world,” *Science*, vol. 334, no. 6054, pp. 308–309, 2011.
- [63] J. M. Pearce, “The case for open source appropriate technology,” *Environment, Development and Sustainability*, vol. 14, no. 3, pp. 425–431, 2012.

- [64] R. Acosta, *Open source hardware*. PhD thesis, Massachusetts Institute of Technology, 2009.
- [65] C. Zhang, N. C. Anzalone, R. P. Faria, and J. M. Pearce, “Open-source 3D-printable optics equipment,” *PloS one*, vol. 8, no. 3, p. e59840, 2013.
- [66] G. C. Anzalone, A. G. Glover, and J. M. Pearce, “Open-source colorimeter,” *Sensors*, vol. 13, no. 4, pp. 5338–5346, 2013.
- [67] “ISO 7027: Water quality — determination of turbidity,” 1999.
- [68] A. C. Ziegler, “Issues related to use of turbidity measurements as a surrogate for suspended sediment,” in *Turbidity and other sediment surrogates workshop*, vol. 1, 2002.
- [69] D. Hongve and G. Åkesson, “Comparison of nephelometric turbidity measurements using wavelengths 400–600 and 860nm,” *Water Research*, vol. 32, no. 10, pp. 3143–3145, 1998.
- [70] M. Bergquist, J. Ljungberg, and B. Rolandsson, “A historical account of the value of free and open source software: From software commune to commercial commons,” in *Open Source Systems: Grounding Research*, pp. 196–207, Springer, 2011.
- [71] K. W. Miller, J. Voas, and T. Costello, “Free and open source software,” *IT professional*, vol. 12, no. 6, pp. 14–16, 2010.
- [72] “94 percent of the world’s top 500 supercomputers run linux.” <https://www.linux.com/news/enterprise/high-performance/147-high-performance/666669-94-percent-of-the-worlds-top-500-supercomputers-run-linux>. (Accessed January 25, 2013).
- [73] D. Rosenberg, “Survey: 98 percent of enterprises using open source.” <http://www.cnet.com/news/survey-98-percent-of-enterprises-using-open-source>, 2010. Accessed January 25, 2013.
- [74] “The open source definition (annotated).” <http://opensource.org/docs/definition.html>. (Accessed January 25, 2013).
- [75] L. H. Glynn, K. A. Hallgren, J. M. Houck, and T. B. Moyers, “CACTI: free, open-source software for the sequential coding of behavioral interactions,” *PloS one*, vol. 7, no. 7, p. e39740, 2012.
- [76] T. Lang, “Advancing global health research through digital technology and sharing data,” *Science*, vol. 331, no. 6018, pp. 714–717, 2011.

- [77] S. Meister, D. M. Plouffe, K. L. Kuhlen, G. M. Bonamy, T. Wu, S. W. Barnes, S. E. Bopp, R. Borboa, A. T. Bright, J. Che, *et al.*, “Imaging of plasmodium liver stages to drive next-generation antimalarial drug discovery,” *Science*, vol. 334, no. 6061, pp. 1372–1377, 2011.
- [78] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas, “TrakEM2 software for neural circuit reconstruction,” *PloS one*, vol. 7, no. 6, p. e38011, 2012.
- [79] K. Kumar, V. Desai, L. Cheng, M. Khitrov, D. Grover, R. V. Satya, C. Yu, N. Zavaljevski, and J. Reifman, “AGeS: a software system for microbial genome sequence annotation,” *PLoS One*, vol. 6, no. 3, p. e17469, 2011.
- [80] W. Christian, F. Esquembre, and L. Barbato, “Open source physics,” *Science*, vol. 334, no. 6059, pp. 1077–1078, 2011.
- [81] T. C. Marzullo and G. J. Gage, “The SpikerBox: a low cost, open-source bioamplifier for increasing public participation in neuroscience inquiry,” *PLoS One*, vol. 7, no. 3, p. e30837, 2012.
- [82] B. Bruns, “Open sourcing nanotechnology research and development: issues and opportunities,” *Nanotechnology*, vol. 12, no. 3, p. 198, 2001.
- [83] U. Mushtaq and J. M. Pearce, “Open source appropriate nanotechnology,” in *Nanotechnology and global sustainability*, pp. 191–213, 2012.
- [84] J. M. Pearce, “Physics: Make nanotechnology research open-source,” *Nature*, vol. 491, no. 7425, pp. 519–521, 2012.
- [85] “NEMA motor - RepRapWiki.” http://reprap.org/wiki/NEMA_Motor. (Accessed December 19, 2013).
- [86] “OpenSCAD.” <http://opencad.org>. (Accessed December 18, 2013).
- [87] “mtu-most/linear-actuator: Code and model files for open source linear actuator.” <https://github.com/mtu-most/linear-actuator>. (Accessed December 19, 2013).
- [88] “Cura user manual.” <http://blog.ultimaker.com/cura-user-manual>. (Accessed December 19, 2013).
- [89] “MOST RepRap build.” http://www.appropedia.org/MOST_RepRap_build. (Accessed December 19, 2013).
- [90] “Delta build overview.” http://www.appropedia.org/Delta_Build_Overview:MOST. (Accessed December 19, 2013).

- [91] “Repetier software.” <http://www.repetier.com>. (Accessed December 19, 2013).
- [92] “Raspbian.” <http://www.raspbian.org>. (Accessed December 19, 2013).
- [93] M. Woelfle, P. Olliaro, and M. H. Todd, “Open science is a research accelerator,” *Nature Chemistry*, vol. 3, pp. 745–748, 2011.
- [94] M. Nielsen, *Reinventing discovery: the new era of networked science*. Princeton University Press, 2012.
- [95] E. Raymond, “The cathedral and the bazaar,” *Knowledge, Technology & Policy*, vol. 12, no. 3, pp. 23–49, 1999.
- [96] K. R. Lakhani and E. Von Hippel, “How open source software works: “free” user-to-user assistance,” *Research policy*, vol. 32, no. 6, pp. 923–943, 2003.
- [97] S. Weber, “The success of open source (vol. 897),” 2004.
- [98] J. Söderberg, *Hacking capitalism: The free and open source software movement*. Routledge, 2008.
- [99] B. Kogut and A. Metiu, “Open-source software development and distributed innovation,” *Oxford Review of Economic Policy*, vol. 17, no. 2, pp. 248–264, 2001.
- [100] “Linux dominates supercomputers as never before.” <http://www.zdnet.com/article/linux-dominates-supercomputers-as-never-before>. (Accessed May 22, 2016).
- [101] A. Mockus, R. T. Fielding, and J. D. Herbsleb, “Two case studies of open source software development: Apache and mozilla,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.
- [102] G. Rundle, *A Revolution in the Making*. Affirm Press, 2014.
- [103] T. Baden, A. M. Chagas, G. J. Gage, T. C. Marzullo, L. L. Prieto-Godino, and T. Euler, “Open labware: 3-D printing your own lab equipment,” *PLoS Biol*, vol. 13, no. 3, p. e1002086, 2015.
- [104] C. D. Kelley, A. Krolick, L. Brunner, A. Burklund, D. Kahn, W. P. Ball, and M. Weber-Shirk, “An affordable open-source turbidimeter,” *Sensors*, vol. 14, no. 4, pp. 7142–7155, 2014.
- [105] M. C. Carvalho and B. D. Eyre, “A low cost, easy to build, portable, and universal autosampler for liquids,” *Methods in Oceanography*, vol. 8, pp. 23–32, 2013.

- [106] E. T. da Costa, M. F. Mora, P. A. Willis, C. L. do Lago, H. Jiao, and C. D. Garcia, “Getting started with open-hardware: Development and control of microfluidic devices,” *Electrophoresis*, vol. 35, no. 16, pp. 2370–2377, 2014.
- [107] T. H. Lücking, F. Sambale, S. Beutel, and T. Scheper, “3D-printed individual labware in biosciences by rapid prototyping: A proof of principle,” *Engineering in Life Sciences*, vol. 15, no. 1, pp. 51–56, 2015.
- [108] B. C. Gross, J. L. Erkal, S. Y. Lockwood, C. Chen, and D. M. Spence, “Evaluation of 3D printing and its potential impact on biotechnology and the chemical sciences,” *Analytical chemistry*, vol. 86, no. 7, pp. 3240–3253, 2014.
- [109] C.-K. Su, S.-C. Hsia, and Y.-C. Sun, “Three-dimensional printed sample load/inject valves enabling online monitoring of extracellular calcium and zinc ions in living rat brains,” *Analytica chimica acta*, vol. 838, pp. 58–63, 2014.
- [110] M. Maldonado-Torres, J. F. López-Hernández, P. Jiménez-Sandoval, and R. Winkler, “‘plug and play’ assembly of a low-temperature plasma ionization mass spectrometry imaging (LTP-MSI) system,” *Journal of proteomics*, vol. 102, pp. 60–65, 2014.
- [111] J. N. Wittbrodt, U. Liebel, and J. Gehrig, “Generation of orientation tools for automated zebrafish screening assays using desktop 3D printing,” *BMC biotechnology*, vol. 14, no. 1, p. 1, 2014.
- [112] D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti, and D. J. Morrow, “The OpenPMU platform for open-source phasor measurements,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 4, pp. 701–709, 2013.
- [113] T. R. Damase, D. Stephens, A. Spencer, and P. B. Allen, “Open source and DIY hardware for DNA nanotechnology labs,” *Journal of biological methods*, vol. 2, no. 3, p. e24, 2015.
- [114] K.-H. Herrmann, C. Gärtner, D. Güllmar, M. Krämer, and J. R. Reichenbach, “3D printing of MRI compatible components: Why every MRI research group should have a low-budget 3D printer,” *Medical engineering & physics*, vol. 36, no. 10, pp. 1373–1380, 2014.
- [115] J. M. Pearce, “Laboratory equipment: Cut costs with open-source hardware,” *Nature*, vol. 505, no. 7485, pp. 618–618, 2014.
- [116] J. M. Pearce, “Quantifying the value of open source hardware development,” *Modern Economy*, vol. 6, no. 1, p. 1, 2015.

- [117] J. Pearce, “Return on investment for open source scientific hardware development,” *Science and Public Policy*, p. scv034, 2016.
- [118] J. S. Cybulski, J. Clements, and M. Prakash, “Foldscope: Origami-based paper microscope,” *PloS one*, vol. 9, no. 6, p. e98781, 2014.
- [119] U. of Cambridge, “The open microscope.” <http://openlabtools.eng.cam.ac.uk/Instruments/Microscope>, 2016. (Accessed May 22, 2016).
- [120] E. Borgen, B. Naume, J. M. Nesland, K. W. Nowels, N. Pavlak, I. Ravkin, and S. Goldbard, “Use of automated microscopy for the detection of disseminated tumor cells in bone marrow samples,” *Cytometry*, vol. 46, no. 4, pp. 215–221, 2001.
- [121] C. A. Holdaway, “Automation of pollen analysis using a computer microscope,” Master’s thesis, Massey University, Turitea, New Zealand, 2004.
- [122] D. N. Mastronarde, “Automated electron microscope tomography using robust prediction of specimen movements,” *Journal of structural biology*, vol. 152, no. 1, pp. 36–51, 2005.
- [123] C. Zhang, B. Wijnen, and J. Pearce, “Open-source 3-D platform for low-cost scientific instrument ecosystem,” *Journal of laboratory automation*, p. 2211068215624406, 2016.
- [124] “Zaber ASR series: Motorized XY microscope stages.” http://www.zaber.com/products/product_group.php?group=ASR. (Accessed May 22, 2016).
- [125] “Leica LMT260 XY scanning stage.” <http://www.leica-microsystems.com/products/stereo-microscopes-microscopes/accessories/details/product/leica-lmt260-xy-scanning-stage>. (Accessed May 22, 2016).
- [126] J. A. Riedl, K. Stouten, H. Ceelie, J. Boonstra, M.-D. Levin, and W. van Gelder, “Interlaboratory reproducibility of blood morphology using the digital microscope,” *Journal of laboratory automation*, vol. 20, no. 6, pp. 670–675, 2015.
- [127] M. V. D’Ambrosio, M. Bakalar, S. Bennuru, C. Reber, A. Skandarajah, L. Nilsson, N. Switz, J. Kamgno, S. Pion, M. Boussinesq, *et al.*, “Point-of-care quantification of blood-borne filarial parasites with a mobile phone microscope,” *Science translational medicine*, vol. 7, no. 286, pp. 286re4–286re4, 2015.
- [128] B. Schneider, G. Vanmeerbeeck, R. Stahl, L. Lagae, J. Dambre, and P. Bienstman, “Neural network for blood cell classification in a holographic microscopy system,” in *Transparent Optical Networks (ICTON), 2015 17th International Conference on*, pp. 1–4, IEEE, 2015.

- [129] A. Carro, M. Perez-Martinez, J. Soriano, D. G. Pisano, and D. Megias, “iM-SRC: converting a standard automated microscope into an intelligent screening platform,” *Scientific reports*, vol. 5, 2015.
- [130] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri, “The ImageJ ecosystem: An open platform for biomedical image analysis,” *Molecular reproduction and development*, vol. 82, no. 7-8, pp. 518–529, 2015.
- [131] “Micro-manager open source microscopy software.” <https://micro-manager.org>. (Accessed May 22, 2016).
- [132] “Hugin — panorama photo stitcher.” <http://hugin.sf.net>. (Accessed May 23, 2016).
- [133] “Fiji is just ImageJ.” <http://fiji.sc>. (Accessed May 23, 2016).
- [134] “Open-source 3-D platform for scientific instruments, koios-master.zip.” <https://osf.io/v2pwa>. (Accessed May 23, 2016).
- [135] “Athena build overview.” http://www.appropedia.org/Athena_Build_Overview. (Accessed May 23, 2016).
- [136] “ASI MS-9400 XY automated stage.” <http://www.asiimaging.com/index.php/products/stages/upright-stages/ms-9400-xy-automated-stage/#tab-id-2>. (Accessed June 24, 2016).
- [137] “High-speed motorized XY scanning stages — MLS203-1 XY stage.” http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=5360. (Accessed June 24, 2016).
- [138] P. Barber, I. Tullis, G. Pierce, R. Newman, J. Prentice, M. Rowley, D. Matthews, S. Ameer-Beg, and B. Vojnovic, “The gray institute ‘open’ high-content, fluorescence lifetime microscopes,” *Journal of microscopy*, vol. 251, no. 2, pp. 154–167, 2013.
- [139] J. P. Sharkey, D. C. Foo, A. Kabla, J. J. Baumberg, and R. W. Bowman, “A one-piece 3D printed microscope and flexure translation stage,” *arXiv preprint arXiv:1509.05394*, 2015.
- [140] “Warwick open source microscope.” <http://wosmic.org>. (Accessed May 23, 2016).
- [141] “RepRap options.” http://reprap.org/wiki/RepRap_Options. (Accessed May 22, 2016).

- [142] T. Wohlers and T. Caffery, *Wohlers Report 2015: Additive Manufacturing and 3D Printing State of the Industry: Annual Worldwide Progress Report*. Wohlers Associates, Fort Collins, 2015.
- [143] “The winners from make:’s digital fabrication shootout.” <http://makezine.com/2015/11/04/digital-fabrication-shootout-winners>. (Accessed May 22, 2016).
- [144] J. Gwamuri, D. Franco, K. Y. Khan, L. Gauchia, and J. M. Pearce, “High-efficiency solar-powered 3-D printers for sustainable development,” *Machines*, vol. 4, no. 1, p. 3, 2016.
- [145] E. Canessa, C. Fonda, and M. Zennaro, *Low-cost 3D printing for science, education and sustainable development*. The Abdus Salam International Centre for Theoretical Physics, 2013.
- [146] C. Mota, “The rise of personal fabrication,” in *Proceedings of the 8th ACM conference on Creativity and cognition*, pp. 279–288, ACM, 2011.
- [147] A. Laplume, G. C. Anzalone, and J. M. Pearce, “Open-source, self-replicating 3-D printer factory for small-business manufacturing,” *The International Journal of Advanced Manufacturing Technology*, pp. 1–10, 2015.
- [148] T. H. Lücking, F. Sambale, B. Schnaars, D. Bulnes-Abundis, S. Beutel, and T. Scheper, “3D-printed individual labware in biosciences by rapid prototyping: In vitro biocompatibility and applications for eukaryotic cell cultures,” *Engineering in Life Sciences*, vol. 15, no. 1, pp. 57–64, 2015.
- [149] J. Pearce, N. Anzalone, and C. Heldt, “Open-source wax RepRap 3-D printer for rapid prototyping paper-based microfluidics,” *Journal of laboratory automation*, p. 2211068215624408, 2016.
- [150] B. Wittbrodt and J. M. Pearce, “Total US cost evaluation of low-weight tension-based photovoltaic flat-roof mounted racking,” *Solar Energy*, vol. 117, pp. 89–98, 2015.
- [151] B. Wittbrodt, J. Laureto, B. Tymrak, and J. M. Pearce, “Distributed manufacturing with 3-D printing: a case study of recreational vehicle solar photovoltaic mounting systems,” *Journal of Frugal Innovation*, vol. 1, no. 1, pp. 1–7, 2015.
- [152] M. Kasparova, L. Grafova, P. Dvorak, T. Dostalova, A. Prochazka, H. Eliasova, J. Prusa, and S. Kakawand, “Possibility of reconstruction of dental plaster cast from 3D digital study models,” *Biomed Eng Online*, vol. 12, no. 49, pp. 1–11, 2013.

- [153] A.-V. Burde, M. Constantiniuc, and R.-S. Câmpian, “Applications of RepRap three-dimensional printers in dentistry—a literature review,” *International Journal of Medical Dentistry*, vol. 5, no. 1, p. 37, 2015.
- [154] G. Niezen, P. Eslambolchilar, and H. Thimbleby, “Open-source hardware for medical devices,” *BMJ innovations*, vol. 2, no. 2, pp. 78–83, 2016.
- [155] M. P. Chae, W. M. Rozen, P. G. McMenamain, M. W. Findlay, R. T. Spychal, and D. J. Hunter-Smith, “Emerging applications of bedside 3D printing in plastic surgery,” *Frontiers in surgery*, vol. 2, 2015.
- [156] B. M. Kuehn, “Clinicians embrace 3D printers to solve unique clinical challenges,” *JAMA*, vol. 315, no. 4, pp. 333–335, 2016.
- [157] J. M. Pearce, “Applications of open source 3-D printing on small farms,” *Organic Farming*, vol. 1, no. 1, pp. 19–35, 2015.
- [158] N. Grujović, M. Radović, V. Kanjevac, J. Borota, G. Grujović, and D. Divac, “3D printing technology in education environment,” in *34th International Conference on Production Engineering*, pp. 29–30, 2011.
- [159] J. Kentzer, B. Koch, M. Thiim, R. W. Jones, and E. Villumsen, “An open source hardware-based mechatronics project: The replicating rapid 3-D printer,” in *Mechatronics (ICOM), 2011 4th International Conference On*, pp. 1–8, IEEE, 2011.
- [160] J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, and M. Abderrahim, “A new open source 3d-printable mobile robotic platform for education,” in *Advances in autonomous mini robots*, pp. 49–62, Springer, 2012.
- [161] C. Schelly, G. C. Anzalone, B. Wijnen, and J. M. Pearce, “Open-source 3-D printing technologies for education: Bringing additive manufacturing to the classroom,” *Journal of Visual Languages & Computing*, vol. 28, pp. 226–237, 2015.
- [162] J. Szulżyk-Cieplak, A. Duda, and B. Sidor, “3D printers—new possibilities in education,” *Advances in Science and Technology Research Journal*, vol. 8, no. 24, 2014.
- [163] S. S. Horowitz and P. H. Schultz, “Printing space: Using 3D printing of digital terrain models in geosciences education and research,” *Journal of Geoscience Education*, vol. 62, no. 1, pp. 138–145, 2014.
- [164] D. B. Short, “Use of 3D printing by museums: Educational exhibits, artifact education, and artifact restoration,” *3D Printing and Additive Manufacturing*, vol. 2, no. 4, pp. 209–215, 2015.

- [165] “Coefficient of linear thermal expansion.” <http://omnexus.specialchem.com/polymer-properties/properties/coefficient-of-linear-thermal-expansion>. (Accessed May 22, 2016).
- [166] B. Stephens, P. Azimi, Z. El Orch, and T. Ramos, “Ultrafine particle emissions from desktop 3D printers,” *Atmospheric Environment*, vol. 79, pp. 334–339, 2013.
- [167] L. T. Sin, A. R. Rahmat, and W. A. Rahman, *Polylactic acid: PLA biopolymer technology and applications*. William Andrew, 2012.
- [168] “The ingeo journey.” <http://www.natureworksllc.com/The-Ingeo-Journey>. (Accessed May 22, 2016).
- [169] “re:3D — life-sized affordable 3D printing.” <https://re3d.org>. (Accessed May 22, 2016).
- [170] D. Kochan, C. C. Kai, and D. Zhaohui, “Rapid prototyping issues in the 21st century,” *Computers in industry*, vol. 39, no. 1, pp. 3–10, 1999.
- [171] P. Mercelis and J.-P. Kruth, “Residual stresses in selective laser sintering and selective laser melting,” *Rapid Prototyping Journal*, vol. 12, no. 5, pp. 254–265, 2006.
- [172] B. Wittbrodt and J. M. Pearce, “The effects of PLA color on material properties of 3-D printed components,” *Additive Manufacturing*, vol. 8, pp. 110–116, 2015.
- [173] T.-M. Wang, J.-T. Xi, and Y. Jin, “A model research for prototype warp deformation in the FDM process,” *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 11-12, pp. 1087–1096, 2007.
- [174] A. Guerrero-de Mier, M. Espinosa, and M. Domínguez, “Bricking: A new slicing method to reduce warping,” *Procedia Engineering*, vol. 132, pp. 126–131, 2015.
- [175] L. Xinhua, L. Shengpeng, L. Zhou, Z. Xianhua, C. Xiaohu, and W. Zhongbin, “An investigation on distortion of PLA thin-plate part in the FDM process,” *The International Journal of Advanced Manufacturing Technology*, vol. 79, no. 5-8, pp. 1117–1126, 2015.
- [176] S. Long and S. Atluri, “A meshless local petrov-galerkin method for solving the bending problem of a thin plate,” *Computer Modeling in Engineering and Sciences*, vol. 3, no. 1, pp. 53–64, 2002.
- [177] D. Garlotta, “A literature review of poly (lactic acid),” *Journal of Polymers and the Environment*, vol. 9, no. 2, pp. 63–84, 2001.

- [178] S. Saeidlou, M. A. Huneault, H. Li, and C. B. Park, “Poly (lactic acid) crystallization,” *Progress in Polymer Science*, vol. 37, no. 12, pp. 1657–1677, 2012.
- [179] D. E. Henton, P. Gruber, J. Lunt, and J. Randall, “Polylactic acid technology,” *Natural fibers, biopolymers, and biocomposites*, vol. 16, pp. 527–577, 2005.
- [180] S. Wu, *Polymer interface and adhesion*. M. Dekker, 1982.
- [181] T. Payne, “Heated build chamber for Rapman 3D printer — instructables.” <http://www.instructables.com/id/Heated-Build-Chamber-for-Rapman-3D-Printer>. (Accessed May 24, 2016).
- [182] “Heated build chamber — RepRap wiki.” http://reprap.org/wiki/Heated_Build_Chamber. (Accessed May 24, 2016).
- [183] “GolemD — RepRap wiki.” <http://reprap.org/wiki/GolemD>. (Accessed May 24, 2016).
- [184] “Introducing kühling & kühling RepRap industrial 3D printer.” <http://www.3ders.org/articles/20130220-introducing-kuhling-kuhling-reprap-industrial-3d-printer.html>. (Accessed May 24, 2016).
- [185] F. Carrasco, P. Pagès, J. Gámez-Pérez, O. Santana, and M. L. MasPOCH, “Processing of poly (lactic acid): characterization of chemical structure, thermal stability and mechanical properties,” *Polymer Degradation and Stability*, vol. 95, no. 2, pp. 116–125, 2010.
- [186] “Ingeo PLA filament in 3D printing.” <http://www.natureworksllc.com/Product-and-Applications/3D-Printing>. (Accessed May 24, 2016).
- [187] “Slic3r — G-code generator for 3D printers.” <http://slic3r.org>. (Accessed May 22, 2016).
- [188] B. Wijnen, “Open science framework — improved model and experimental validation of deformation in fused filament fabrication of poly lactic acid.” <https://osf.io/4kvwm>. (Accessed May 24, 2016).
- [189] “GNU image manipulation program.” <http://www.gimp.org>. (Accessed May 22, 2016).
- [190] “Thingiverse — digital designs for physical objects.” <http://www.thingiverse.com>. (Accessed May 24, 2016).
- [191] “Github — how people build software.” <https://github.com>. (Accessed May 24, 2016).

Appendix A

Franklin Source Statistics

This appendix contains a list of source code files of Franklin, and the number of lines that they contain. Another interesting part would have been the amount of development time. However, there is no objective conversion possible from lines of code to hours of work, and this information was not recorded. The files are all available on Github [18].

Lines	Filename	Function
13	.gitignore	workflow enhancement
661	LICENSE	legal matter
88	Makefile	build rules
28	README.md	documentation
140	calibrate/Athena.ini	initial settings
45	calibrate/bed-map	automated calibration
12	calibrate/calibration.tex	documentation
274	calibrate/delta-calibrate	automated calibration
57	calibrate/delta.ini	initial settings
31	calibrate/export	profile management
45	calibrate/import	profile management
56	calibrate/melzi.ini	initial settings
661	calibrate/printer3d-calibrate	automated calibration
168	calibrate/printer3d-calibrate.gui.in	automated calibration
79	calibrate/prusa.ini	initial settings
229	calibrate/rambo.ini	initial settings
69	calibrate/ramps.ini	initial settings
434	calibrate/reset.ini	initial settings

Lines	Filename	Function
11	debian/changelog	build rules
1	debian/compat	build rules
21	debian/control	build rules
682	debian/copyright	build rules
8	debian/franklin.default	build rules
4	debian/franklin.dirs	build rules
170	debian/franklin.init	build rules
17	debian/franklin.install	build rules
3	debian/franklin.links	build rules
1	debian/franklin.manpages	build rules
1	debian/franklin.mime	build rules
29	debian/franklin.postinst	build rules
10	debian/franklin.postrm	build rules
6	debian/franklin.udev	build rules
38	debian/rules	build rules
1	debian/source/format	build rules
7	doc/examples/README.md	documentation
10	doc/parity.txt	documentation
41	doc/plan.txt	documentation
96	doc/protocol.txt	documentation
90	firmware/Makefile	build rules
1226	firmware/arch-avr.h	AVR specific code
221	firmware/arch-bbb.h	BeagleBone specific code
224	firmware/arch-sim.h	simulator specific code
542	firmware/firmware.h	declarations
206	firmware/firmware.ino	main loop
623	firmware/packet.cpp	incoming packet handling
579	firmware/serial.cpp	serial communication handling
89	firmware/setup.cpp	firmware initialization
131	firmware/timer.cpp	recurring event checks
3	firmware/vg	valgrind helper for simulator target
4	server/bb/Makefile	build rules
36	server/bb/avrdude.conf	programmer hardware definition
47	server/bb/flash-bb-0	programmer for ttyS0
45	server/bb/flash-bb-4	programmer for ttyS4
53	server/bb/setpin.c	change pin function

Lines	Filename	Function
77	server/cdriver/Makefile	build rules
1434	server/cdriver/arch-avr.h	AVR specific code
703	server/cdriver/arch-bbb.h	BeagleBone specific code
68	server/cdriver/athena-0.dts	BeagleBone device tree
94	server/cdriver/base.cpp	cdriver main loop
136	server/cdriver/bbb_pru.asm	BeagleBone PRU assembly
618	server/cdriver/cdriver.h	declarations
40	server/cdriver/configuration.h	build time configuration
138	server/cdriver/debug.cpp	debugging helpers
153	server/cdriver/globals.cpp	global settings
72	server/cdriver/gpio.cpp	gpio settings
70	server/cdriver/hostserial.cpp	communications with Python driver
550	server/cdriver/move.cpp	movement planning
752	server/cdriver/packet.cpp	handling packets from Python driver
482	server/cdriver/run.cpp	stored G-Code handling
655	server/cdriver/serial.cpp	serial communication handling
147	server/cdriver/setup.cpp	initialization
920	server/cdriver/space.cpp	space settings
58	server/cdriver/storage.cpp	settings storage helpers
273	server/cdriver/temp.cpp	temp settings
293	server/cdriver/type-cartesian.cpp	cartesian and related details
224	server/cdriver/type-delta.cpp	delta details
129	server/cdriver/type-polar.cpp	polar details
3	server/cdriver/vg	valgrind debugging helper
67	server/control.py	serial port autodetection helper
3275	server/driver.py	Python driver
7	server/firmware/Makefile	build rules
59	server/franklin.8.ronn	documentation
19	server/html/Makefile	build rules
44	server/html/index.html	web interface
371	server/html/main.css	web interface
2023	server/html/main.js	web interface
430	server/html/server.js	communication helpers
916	server/html/templates.js	component widgets
160	server/protocol.py	communication definitions
950	server/server.py	server

Lines	Filename	Function
32	util/Makefile	build rules
224	util/c457a-ui	microscope tool
43	util/c457a-ui.gui.in	microscope tool interface
88	util/display	microscope tool interface
96	util/embroidery.py	embroidery tool
223	util/franklin_joystick.py	joystick helper
52	util/joystick.c	joystick helper
25534	total	